Andrzej Bargiela and Witold Pedrycz (Eds.)

# Human-Centric Information Processing Through Granular Modelling

Studies in Computational Intelligence, Volume 182

**Editor-in-Chief**
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
*E-mail:* kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 159. Dimitri Plemenos and Georgios Miaoulis (Eds.)
*Artificial Intelligence Techniques for Computer Graphics,* 2009
ISBN 978-3-540-85127-1

Vol. 160. P. Rajasekaran and Vasantha Kalyani David
*Pattern Recognition using Neural and Functional Networks,* 2009
ISBN 978-3-540-85129-5

Vol. 161. Francisco Baptista Pereira and Jorge Tavares (Eds.)
*Bio-inspired Algorithms for the Vehicle Routing Problem,* 2009
ISBN 978-3-540-85151-6

Vol. 162. Costin Badica, Giuseppe Mangioni,
Vincenza Carchiolo and Dumitru Dan Burdescu (Eds.)
*Intelligent Distributed Computing, Systems and Applications,* 2008
ISBN 978-3-540-85256-8

Vol. 163. Pawel Delimata, Mikhail Ju. Moshkov,
Andrzej Skowron and Zbigniew Suraj
*Inhibitory Rules in Data Analysis,* 2009
ISBN 978-3-540-85637-5

Vol. 164. Nadia Nedjah, Luiza de Macedo Mourelle,
Janusz Kacprzyk, Felipe M.G. França
and Alberto Ferreira de Souza (Eds.)
*Intelligent Text Categorization and Clustering,* 2009
ISBN 978-3-540-85643-6

Vol. 165. Djamel A. Zighed, Shusaku Tsumoto,
Zbigniew W. Ras and Hakim Hacid (Eds.)
*Mining Complex Data,* 2009
ISBN 978-3-540-88066-0

Vol. 166. Constantinos Koutsojannis and Spiros Sirmakessis (Eds.)
*Tools and Applications with Artificial Intelligence,* 2009
ISBN 978-3-540-88068-4

Vol. 167. Ngoc Thanh Nguyen and Lakhmi C. Jain (Eds.)
*Intelligent Agents in the Evolution of Web and Applications,* 2009
ISBN 978-3-540-88070-7

Vol. 168. Andreas Tolk and Lakhmi C. Jain (Eds.)
*Complex Systems in Knowledge-based Environments: Theory, Models and Applications,* 2009
ISBN 978-3-540-88074-5

Vol. 169. Nadia Nedjah, Luiza de Macedo Mourelle and
Janusz Kacprzyk (Eds.)
*Innovative Applications in Data Mining,* 2009
ISBN 978-3-540-88044-8

Vol. 170. Lakhmi C. Jain and Ngoc Thanh Nguyen (Eds.)
*Knowledge Processing and Decision Making in Agent-Based Systems,* 2009
ISBN 978-3-540-88048-6

Vol. 171. Chi-Keong Goh, Yew-Soon Ong and Kay Chen Tan (Eds.)
*Multi-Objective Memetic Algorithms,* 2009
ISBN 978-3-540-88050-9

Vol. 172. I-Hsien Ting and Hui-Ju Wu (Eds.)
*Web Mining Applications in E-Commerce and E-Services,* 2009
ISBN 978-3-540-88080-6

Vol. 173. Tobias Grosche
*Computational Intelligence in Integrated Airline Scheduling,* 2009
ISBN 978-3-540-89886-3

Vol. 174. Ajith Abraham, Rafael Falcón and Rafael Bello (Eds.)
*Rough Set Theory: A True Landmark in Data Analysis,* 2009
ISBN 978-3-540-89886-3

Vol. 175. Godfrey C. Onwubolu and Donald Davendra (Eds.)
*Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization,* 2009
ISBN 978-3-540-92150-9

Vol. 176. Beniamino Murgante, Giuseppe Borruso and
Alessandra Lapucci (Eds.)
*Geocomputation and Urban Planning,* 2009
ISBN 978-3-540-89929-7

Vol. 177. Dikai Liu, Lingfeng Wang and Kay Chen Tan (Eds.)
*Design and Control of Intelligent Robotic Systems,* 2009
ISBN 978-3-540-89932-7

Vol. 178. Swagatam Das, Ajith Abraham and Amit Konar
*Metaheuristic Clustering,* 2009
ISBN 978-3-540-92172-1

Vol. 179. Mircea Gh. Negoita and Sorin Hintea
*Bio-Inspired Technologies for the Hardware of Adaptive Systems,* 2009
ISBN 978-3-540-76994-1

Vol. 180. Wojciech Mitkowski and Janusz Kacprzyk (Eds.)
*Modelling Dynamics in Processes and Systems,* 2009
ISBN 978-3-540-92202-5

Vol. 181. Georgios Miaoulis and Dimitri Plemenos (Eds.)
*Intelligent Scene Modelling Information Systems,* 2009
ISBN 978-3-540-92901-7

Vol. 182. Andrzej Bargiela and Witold Pedrycz (Eds.)
*Human-Centric Information Processing Through Granular Modelling,* 2009
ISBN 978-3-540-92915-4

Andrzej Bargiela
Witold Pedrycz
(Eds.)

# Human-Centric Information Processing Through Granular Modelling

Springer

Andrzej Bargiela
School of Computer Science
The University of Nottingham
Malaysia Campus, Jalan Broga
43500 Semenyih
Selangor Darul Ehsan
Malaysia
E-mail: andrzej.bargiela@nottingham.edu.my

Witold Pedrycz
Department of Electrical & Computer Engineering
University of Alberta
Edmonton AB T6R 2G7
Canada
and
System Research Institute
Polish Academy of Sciences
Warsaw, Poland
E-mail: pedrycz@ee.ualberta.ca

ISBN 978-3-540-92915-4 e-ISBN 978-3-540-92916-1

DOI 10.1007/978-3-540-92916-1

Studies in Computational Intelligence ISSN 1860949X

Library of Congress Control Number: 2008942137

*Typeset & Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

# Preface

Information granules and their processing permeate a way in which we perceive the world, carryout processing at the conceptual (abstract) level, and communicate our findings to the surrounding environment. The importance of information granulation becomes even more apparent when we are faced with a rapidly growing flood of data, become challenged to make decisions in complex data settings and are required to appreciate the context from which the data is derived. Human centricity of systems that claim to be "intelligent" and the granular computing come hand in hand. It is not surprising at all to witness that the paradigm of Granular Computing has started to gain visibility and continues along this path by gathering interest from the circles of academics and practitioners. It is quite remarkable that the spectrum of application and research areas that have adopted information granulation as a successful strategy for dealing with information complexity covers such diverse fields as bioinformatics, image understanding, environmental monitoring, urban sustainability, to mention few most visible in the literature.

Undoubtedly, there are two important aspects of Granular Computing that are worth stressing. First, there are several formalisms in which information granules are articulated so be intervals (sets), fuzzy sets, rough sets, soft sets, approximate sets, near sets and alike. They are complementary and each of them offers some interesting views at the complexity of the world and cyberspace. All of them are the key players of Granular Computing by enriching the conceptual, methodological and algorithmic landscape of the area. While there are some signs of vital synergy (manifesting through e.g., hybrid constructs such as rough fuzzy sets, fuzzy rough sets and alike), one has to admit that there is a long way to go before we reach a point of a fully coherent and unified theory of information granules, information granulation and processing of such constructs. In this sense, the period of growth we are witnessing today is the most exciting stage of the development of the new computing paradigm. Second, the granularity of information leads us immediately to a hierarchy of concepts, models and associated computing machinery. A suitable selection of the level of granularity is crucial to the effective realization of all cognitive and computing faculties. The computational framework for processing information granules is likely to evolve alongside the algorithmic developments of information granulation and granular modeling. Novel, ubiquitous or biologically inspired computational machinery is actively researched in the form of simulated environments and is likely to lead to physical hardware implementations upon successful demonstration of granular information processing potential.

This volume is fully reflective of the diversity of Granular Computing both in terms of the underlying methodology as well as algorithms and applications. There is a well-delineated group of contributions (Nguen and Skowron; Yao; Peters and Ramanna; Zhang et al. ) which embarks on the fundamentals of Granular Computing by casting them in a new and illuminating perspective. The issues of constructing information granules have always been one of the focal points of Granular Computing. The contributions authored by Mencar, Fazendeiro and Valente de Oliveira emphasize the aspect of interpretability of information granules no matter in which way they were formed. Fuzzy sets and their generalizations such as Type-2 fuzzy sets have occupied a visible position in Granular Computing. Interestingly, there are a number of contributions (Gottwald; John and Coupland; Castillo and Melin) which elaborate on the concepts and algorithmic machinery of fuzzy sets. The applied end of the Granular Computing is fully reflected in the volume through a series of papers ( Apolloni, Bassis, and Zippo; Acampora, Loia, and Vasilakos; Pizzi; Das and Mitra; Burczynski and Orantek; Sawase, Nobuhara and Bede) – those contributions are a convincing testimony to the diversity of applications of Granular Computing. The contribution from Greensmith and Aickelin is a representative of the growing research trend into bio-inspired computing environments for granular information processing.

Our sincere gratitude goes to the authors who enthusiastically responded to our project and offered their expertise and shared their recent research findings with the research community. The reviewers, who provided objective and sometimes critical but always constructive comments, played an important role is shaping up the book– we are definitely grateful for all the input we received from them during the realization of this project.

We definitely enjoyed working on this volume by seeing how new ideas come into existence. It is our hope that the readers will find this volume both intellectually stimulating in their research and practical pursuits.

We would like to express our gratitude to the Engineering and Physical Sciences Research Council (EPSRC) for their generous support provided to realize the project and their long-term support of research excellence.

November 2008

Andrzej Bargiela
Witold Pedrycz

# Contents

# Rough-Granular Computing in Human-Centric Information Processing

Tuan Trung Nguyen and Andrzej Skowron

Polish-Japanese Institute of Information Technology, Koszykowa 86,
02-008 Warsaw, Poland
nttrung@pjwstk.edu.pl
Institute of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland
skowron@mimuw.edu.pl

**Abstract.** In ubiquitous computing, users are expected to continuously interact with computing devices, to suggest strategies and hypotheses, to pass over new facts from domain knowledge, to explain untypical cases in dialogs with the devices, etc. These devices therefore need to, at least in an approximate sense, understand the compound, vague concepts used by humans. We discuss current results and research directions on the approximation of compound vague concepts, which are based on rough-granular computing. In particular, we use hierarchical methods for the approximation of domain ontologies of vague concepts. We also discuss an extension of the proposed approach for approximate reasoning about interactive computations performed on complex granules by systems of agents in dynamically changing environments.

## 1 Selected Basic Issues on Granular Computing

In this section, we discuss some basic issue of *Granular Computing* (GC). We consider granules as constructive definitions of sets used in assembling objects satisfying a given specification at least to satisfactory degree. Granules are usually defined by granule systems [51, 52] in which some elementary granules are distinguished, together with operations making it possible to define new granules from these elementary granules, or from already defined granules. Among special types of operations on granules, one can distinguish the fusion and decomposition operations. For more readings on GC, the reader is referred to [2, 35, 38, 39, 42].

### 1.1 Synthesis of Complex Objects Satisfying Vague Specifications

One of the central issues related to granules is the definition of inclusion relations and closeness relations (measures). The concept of rough inclusion from rough mereology [45] can be used as a starting point in searching for constructive measures of inclusion or closeness of granules. Note that these measures should be defined for granules with different complexity structures.

In real-life applications, we often deal with problems where not only is the information about objects partial, but also the specification of problems is written in natural language. Hence, such specifications involve vague or/and imperfect

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 1–30.
springerlink.com

concepts. Problems we are trying to solve can be characterized as searching for complex objects satisfying a given specification to a satisfactory degree [45]. These complex objects should be synthesized from more elementary ones using available operations. Moreover, usually only partial information about these objects and concepts used in the specifications are available.

In the following section, we discuss searching for relevant granules as a kind of optimization problem in GC.

### 1.2 Optimization in Discovery of Compound Granules

This section is based on the approach discussed in [20, 30].

The problem considered in this section is the evaluation of perception as a means of optimizing various tasks. The solution to this problem hearkens back to early research on rough set theory and approximation. For example, in 1982, Ewa Orłowska observed that approximation spaces serve as a formal counterpart of perception.

In this chapter, the evaluation of perception is at the level of approximation spaces. The quality of an approximation space relative to a given approximated set of objects is a function of the description length of an approximation of the set of objects and the approximation quality of this set. In granular computing (GC), the focus is on discovering granules satisfying selected criteria. These criteria take inspiration from the minimal description length (MDL) principle proposed by Jorma Rissanen in 1983. In this section, the role of approximation spaces in modeling compound granules satisfying such criteria is discussed.

First, we recall the definition of an approximation space from [50]. Approximation spaces can be treated as granules used for concept approximation. They are examples of special parameterized relational structures. Tuning parameters make it possible to search for relevant approximation spaces relative to given concepts.

**Definition 1.** *A parameterized approximation space is a system $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$, where*

- *$U$ is a non-empty set of objects,*
- *$I_{\#} : U \rightarrow P(U)$ is an uncertainty function, where $P(U)$ denotes the power set of $U$,*
- *$\nu_{\$} : P(U) \times P(U) \rightarrow [0, 1]$ is a rough inclusion function,*

*and $\#, \$$ denote vectors of parameters (the indexes $\#, \$$ will be omitted if it does not lead to misunderstanding).*

The uncertainty function defines for every object $x$, a set of objects described similarly to $x$. The set $I(x)$ is called the neighborhood of $x$ (see, *e.g.*, [36, 50]).

The rough inclusion function $\nu_{\$} : P(U) \times P(U) \rightarrow [0, 1]$ defines the degree of inclusion of $X$ in $Y$, where $X, Y \subseteq U$.

In the simplest case it can be defined by (see, *e.g.*, [50, 36]):

$$\nu_{SRI}(X, Y) = \begin{cases} \frac{card(X \cap Y)}{card(X)}, & \text{if } X \neq \emptyset, \\ 1, & \text{if } X = \emptyset. \end{cases}$$

The lower and the upper approximations of subsets of $U$ are defined as follows.

**Definition 2.** *For any approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ and any subset $X \subseteq U$, the lower and upper approximations are defined by*
$LOW\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}\left(x\right), X\right) = 1\},$
$UPP\left(AS_{\#,\$}, X\right) = \{x \in U : \nu_{\$}\left(I_{\#}\left(x\right), X\right) > 0\}$, *respectively.*

The lower approximation of a set $X$ with respect to the approximation space $AS_{\#,\$}$ is the set of all objects that can be classified with certainty as objects of $X$ with respect to $AS_{\#,\$}$. The upper approximation of a set $X$ with respect to the approximation space $AS_{\#,\$}$ is the set of all objects which can be possibly classified as objects of $X$ with respect to $AS_{\#,\$}$.

Several known approaches to concept approximation can be covered using this approach to approximation spaces (see, *e.g.*, references in [50]). For more details on approximation spaces, the reader is referred to, *e.g.*, [10, 37, 53, 41, 44].

A key task in granular computing is the information granulation process that leads to the formation of information aggregates (with inherent patterns) from a set of available objects. A methodological and algorithmic issue is the formation of transparent (understandable) information granules inasmuch as they should provide a clear and understandable description of patterns present in sample objects [2, 39]. Such a fundamental property can be formalized by a set of constraints that must be satisfied during the information granulation process. Usefulness of these constraints is measured by the quality of an approximation space:

$$Quality_1 : Set\_AS \times P(U) \rightarrow [0, 1],$$

where $U$ is a non-empty set of objects and $Set\_AS$ is a set of possible approximation spaces with the universe $U$.

*Example 1.* If $UPP(AS, X)) \neq \emptyset$ for $AS \in Set\_AS$ and $X \subseteq U$ then

$$Quality_1(AS, X) = \nu_{SRI}(UPP(AS, X), LOW(AS, X)) = \frac{card(LOW(AS, X))}{card(UPP(AS, X))}.$$

The value $1 - Quality_1(AS, X)$ expresses the degree of completeness of our knowledge about $X$, given the approximation space $AS$.

*Example 2.* In applications, we usually use another quality measure analogous to the minimal length principle [47, 56] where also the description length of approximation is included. Let us denote by $description(AS, X)$ the description length of approximation of $X$ in $AS$. The description length may be measured, *e.g.*, by the sum of description lengths of algorithms testing membership for neighborhoods used in construction of the lower approximation, the upper approximation, and the boundary region of the set $X$. Then the quality $Quality_2(AS, X)$ can be defined by

$$Quality_2(AS, X) = g(Quality_1(AS, X), description(AS, X)),$$

where $g$ is a relevant function used for fusion of values $Quality_1(AS, X)$ and $description(AS, X)$. This function $g$, for instance, may involve weights assigned by experts to both criteria.

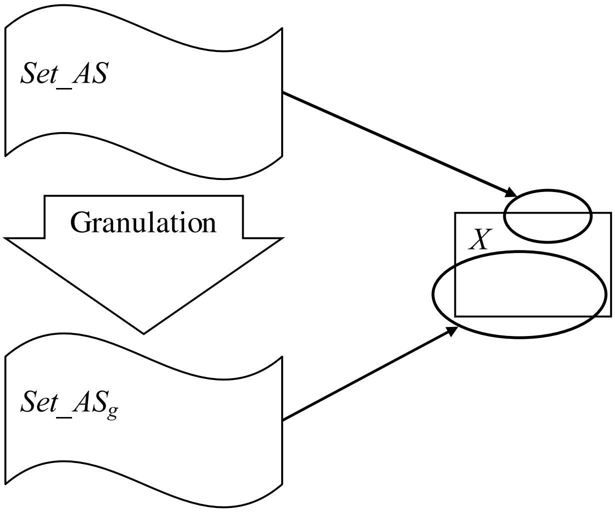


**Fig. 1.** Granulation of parameterized approximation spaces

One can consider different optimization problems relative to a given class *Set_AS* of approximation spaces. For example, for a given $X \subseteq U$ and a threshold $t \in [0,1]$, one can search for an approximation space $AS$ satisfying the constraint $Quality_2(AS, X) \geq t$.

Another example involves searching for an approximation space satisfying additionally the constraint $Cost(AS) < c$ where $Cost(AS)$ denotes the cost of an approximation space $AS$ (*e.g.*, measured by the number of attributes used to define neighborhoods in $AS$) and $c$ is a given threshold. In the following example, we consider also costs of searching for relevant approximation spaces in a given family defined by a parameterized approximation space (see Figure 1). Any parameterized approximation space $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$ is a family of approximation spaces. The cost of searching in such a family for a relevant approximation space for a given concept $X$ approximation can be treated as a factor of the quality measure of approximation of $X$ in $AS_{\#,\$} = (U, I_{\#}, \nu_{\$})$. Hence, such a quality measure of approximation of $X$ in $AS_{\#,\$}$ can be defined by

$$Quality_3(AS_{\#,\$}, X) = h(Quality_2(AS, X), Cost\_Search(AS_{\#,\$}, X)),$$

where $AS$ is the result of searching in $AS_{\#,\$}$, $Cost\_Search(AS_{\#,\$}, X)$ is the cost of searching in $AS_{\#,\$}$ for $AS$, and $h$ is a fusion function, *e.g.*, assuming that the values of $Quality_2(AS, X)$ and $Cost\_Search(AS_{\#,\$}, X)$ are normalized to interval $[0,1]$ $h$ could be defined by a linear combination of $Quality_2(AS, X)$ and $Cost\_Search(AS_{\#,\$}, X)$ of the form

$$\lambda Quality_2(AS, X) + (1 - \lambda) Cost\_Search(AS_{\#,\$}, X),$$

where $0 \leq \lambda \leq 1$ is a weight measuring an importance of quality and cost in their fusion.

We assume that the fusion functions $g, h$ in the definitions of quality are monotonic relative to each argument.

Let $AS \in Set\_AS$ be an approximation space relevant for approximation of $X \subseteq U$, *i.e.*, $AS$ is the optimal (or semi-optimal) relative to $Quality_2$. By $Granulation(AS_{\#,\$})$ we denote a new parameterized approximation space obtained by granulation of $AS_{\#,\$}$. For example, $Granulation(AS_{\#,\$})$ can be obtained by reducing the number of attributes or inclusion degrees (*i.e.*, possible values of the inclusion function). Let $AS'$ be an approximation space in $Granulation(AS_{\#,\$})$ obtained as the result of searching for optimal (semi-optimal) approximation space in $Granulation(AS_{\#,\$})$ for approximation of $X$.

We assume that three conditions are satisfied:

- after granulation of $AS_{\#,\$}$ to $Granulation(AS_{\#,\$})$ the following property holds: the cost
$$Cost\_Search(Granulation(AS_{\#,\$}), X),$$
is much lower than the cost $Cost\_Search(AS_{\#,\$}, X)$;
- The $description(AS', X)$ is much shorter than the $description(AS, X)$, *i.e.*, the description length of $X$ in the approximation space $AS'$ is much shorter than the description length of $X$ in the approximation space $AS$;
- $Quality_1(AS, X)$ and $Quality_1(AS', X)$ are sufficiently close.

The last two conditions should guarantee that the values $Quality_2(AS, X)$ and $Quality_2(AS', X)$ are comparable and this condition together with the first condition about the cost of searching should assure that

$$Quality_3(Granulation(AS_{\#,\$}, X)) \text{ is much better than } Quality_3(AS_{\#,\$}, X).$$

Taking into account that parameterized approximation spaces are examples of parameterized granules, one can generalize the above example of parameterized approximation space granulation to the case of granulation of parameterized granules.

In the process of searching for (sub-)optimal approximation spaces, different strategies may be used. Let us consider an example of such strategies [55]. In the example, $DT = (U, A, d)$ denotes a decision system (a given sample of data), where $U$ is a set of objects, $A$ is a set of attributes and $d$ is a decision. We assume that for any object $x$, only partial information, equal to the $A$-signature of $x$ (object signature, for short), is available, *i.e.*, $Inf_A(x) = \{(a, a(x)) : a \in A\}$ and analogously for any concept, only partial information about this concept by a sample of objects is provided, *e.g.*, in the form of decision table. One can use object signatures as new objects in a new relational structure $\mathcal{R}$. In this relational structure $\mathcal{R}$ some relations between object signatures are also modelled, *e.g.*, defined by the similarities of these object signatures. Discovery of relevant relations on object signatures is an important step in searching for relevant approximation spaces. In this way, a class of relational structures representing perception of objects and their parts is constructed. In the next step, we select a language $\mathcal{L}$ of formulas expressing properties over the defined relational structures and we search for relevant formulas in $\mathcal{L}$. The semantics of formulas (*e.g.*, with one free variable) from $\mathcal{L}$ are subsets of object signatures. Observe that each object signature defines a neighborhood of objects from a given sample (*e.g.*, decision table $DT$) and another set on the whole universe of objects being an extension of $U$. Thus, each formula from $\mathcal{L}$ defines a family of sets of objects over the sample and also another family of sets over the universe of all objects. One can use such families can to define new neighborhoods of a new approximation space, *e.g.*, by taking their unions. In the searching process for relevant neighborhoods, we use information encoded in the given sample. More relevant neighborhoods make it possible to define relevant approximation spaces (from the point of view of the optimization criterion). It is worth to mention that often this searching process is even more sophisticated. For example, one can discover several relational structures (e.g., corresponding to different attributes) and formulas over such structures defining different families of neighborhoods from the original approximation space. Next such families of neighborhoods can be merged into neighborhoods in a new approximation space. This kind of modeling is typical for hierarchical modeling [8], *e.g.*, when we search for a relevant approximation space for objects composed from parts for which some relevant approximation spaces have been already found.

## 2 Granular Computing and Human Perception: Learning in Dialog with Human Experts

The hierarchical learning approach takes advantage of additional domain knowledge provided by human experts. In order to best employ this knowledge, it relies on the observation that human thinking and perception in general, and their reasoning while performing classification tasks in particular, can:

- inherently comprise different levels of abstraction,
- display a natural ability to switch focus from one level to another,
- operate on several levels simultaneously.

Such processes are natural subjects for the *Granular Computing* paradigm, which encompasses theories, methods, techniques and tools for such fields as problem solving, information processing, human perception evaluation, analysis of complex systems and many others. It is built around the concept of *information granules*, which can be understood as collections of *values that are drawn together by indistinguishability, equivalence, similarity, or proximity* [63]. Granular Computing follows the human ability to perceive things in different levels of abstraction (*granularity*), to concentrate on a particular level of interest while preserving the ability to instantly switch to another level in case of need. This allows to obtain different levels of knowledge and, which is important, a better understanding of the inherent structure of this knowledge.

The concept of information granules is closely related to the imprecise nature of human reasoning and perception. Granular Computing therefore provides excellent tools and methodologies for problems involving flexible operations on imprecise or approximated concepts expressed in natural language.

One of the possible approaches in developing methods for compound concept approximations can be based on the layered (hierarchical) learning [11, 57]. Inducing concept approximation should be developed hierarchically starting from concepts that can be directly approximated using sensor measurements toward compound target concepts related to perception. This general idea can be realized using additional domain knowledge represented in natural language. For example, one can use some rules of behavior on the roads, expressed in natural language, to assess from recordings (made, e.g., by camera and other sensors) of actual traffic situations, if a particular situation is safe or not (see, e.g., [8, 9, 14, 31]). The hierarchical learning has been also used for identification of risk patterns in medical data and extended for therapy planning (see, e.g. [6, 7]). Another application of hierarchical learning for sunspot classification is reported in [33]. To deal with such problems one should develop methods for concept approximations together with methods aiming at approximation of reasoning schemes (over such concepts) expressed in natural language. The foundations of such an approach, creating a core of perception logic, are based on rough set theory [14, 36, 37] and its extension rough mereology [35, 45, 51]. The (approximate) Boolean reasoning methods can be scaled to the case of compound concept approximation.

Let us observe that hierarchical modeling employs some general mechanisms emphasized in [22] dealing with a kind of “interplay” between syntax and semantics. The key observation is that the syntax on one level is used to define semantical structures (or their clusters) on the next level of hierarchy. One can interpret them in the framework of the Bairwise classifications [4] as operations on such classifications or as a kind of sums of information systems [54]. They allow us gradually to model structures of granules representing “wider” context of perceived objects. In this way, it is possible to construct more compound granules interpreted, e.g., as patterns representing properties of, e.g., time windows of states, sequences of such time windows, sets of such sequences, etc.

### 2.1 Hierarchical Modeling and Dealing with Ill-Posed Problems: Toward Generalization of the Minimal Length Principle to the Case of Concept Ontology

As pointed out in [61], machine learning problems can be considered as inverse problems, and in a broad view,

$$A(f) = d,$$

where $A$ can be understood as a model for a phenomena, $f \in \mathcal{F}$ represents a function of some of the model's causal factors, chosen from a class $\mathcal{F}$ of candidate functions, and $d$ denotes some actual observation data pertaining to the phenomena, are generally ill-posed, which means the solution $f$ might not exist, might not be unique, and most importantly, might not be stable. Namely, with a small deviation $\delta$ in the output data $d_\delta$, we have

$$R_\delta(f) = \|A(f) - d_\delta\|, \tag{1}$$

not tending to zero even if $\delta$ tends to zero, where $\| \cdot \|$ is any divergence metrics appropriate for $f$, meaning arbitrarily small deviations in data may cause large deviations in solutions.

One can also give another interpretation of the equation (1). The operator $A$ can be interpreted as a (vague) specification (constraints) of the problem and the goal is to find a solution $f$ satisfying the specification to a satisfactory degree. This satisfactory degree is expressed in (1) by means of the norm. Note that, very often, while dealing with real-life problems we have only a vague specification $A$ rather than a crisp operator $A$. Moreover, due to the uncertainty in specification of $A$ and $f$ the quality measures often can only be estimated from available data. In consequence, one can hardly expect that the relevant measures would be expressed in well known spaces with norms as in (1). In such cases one should look for some other avenues to express, e.g., the phrase *a solution should satisfy a given specification to satisfactory degree* [30].

For dealing with ill posed problems the regularization theory was proposed. The idea of regularization is due to Tikhonov (1963, see [60]). Instead of the equation (1) the following one is considered:

$$R_{\delta,\gamma}(f) = \|A(f) - d_\delta\| + \gamma W(f), \tag{2}$$

where $W(f)$ is a functional measuring the "simplicity" of the solution $f$ and $\gamma$ is a parameter (adjustable in the learning process).

Now, in the equation (2) we have a sum of two arguments. The first one expresses the quality of the solution $f$ and the second one expresses, in a sense, the description length of the solution, using the terminology related to the minimal length principle. For a given parameter $\gamma$ we are searching for $f$ by minimizing the value of $R_{\delta,\gamma}(f)$. By choosing different values of $\gamma$ we may alter our priority given to the first or the second summand of the sum in (2).

Fundamental pattern recognitions problems such as class probability density function estimation from a wide set of potential densities, or parametric estimation of optimal feature subsets, are ill-posed.

On the other hand, if the model $A$ can be decomposed into a combination of simpler sub-models $A_i$, e.g. those involving search spaces with lower Vapnik-Chervonenkis (VC) dimensions, or those for which respective stable sub-solutions $f_i$ can be found inexpensively, chances are that we'll be able to assemble a solution $f$ from sub-solutions $f_i$, which will be better than a solution computed in an all-out attempt for the original problem. However, the challenge in this approach is that there is no known automatic method for the computation of effective decompositions of $A$.

In the hierarchical learning approach, we assume that the decomposition scheme will be provided by an external human expert in an interactive process. Knowledge acquired from human expert will serve as guidance to break the original model $A$ into simpler, more manageable sub-models $A_i$, organized in a lattice-like hierarchy. They would correspond to subsequent levels of abstractions in the hierarchy of perception and reasoning of the human expert.

The mentioned above decomposition should lead to submodels $A_i$ together with pertaining functionals $W_i$ as well as parameters $\gamma_i$. The global optimization criteria become more compound in the decomposition case and should be obtained by fusion of those for submodels. For example, one could assume the following optimization criterion:

$$R^*_{\delta,\gamma}(f) = \sum_i \|A_i(f_i) - d_{\delta_i}\| + \gamma_i W_i(f_i), \tag{3}$$

where the sum is taken over all decomposition submodels and $f$ is the solution corresponding to the root level of decomposition (i.e., to the model $A$)[1]. However, the linear fusion in (3) may be too simplistic for real-life problems, where it is important to learn from data approximations of optimization criteria [25, 30].

### 2.2 Narrowing the Potential Search Space

As stated in [61], the problem of estimating $f$ from a large set $\mathcal{F}$ of possible candidate solutions is ill-posed. One way to alleviate this problem is to employ the so-called Structural Risk Minimization (SRM) technique. The technique, in short, is based on a theorem on the risk's bounds, which essentially states that

$$R(\alpha) \leq R_{emp}(\alpha) + CI(\alpha),$$

which means the risk functional $R(\alpha)$, expressing how far we are from the desired solution for a parameter $\alpha$ from a general parameter set $S$, is bounded by the sum of the empirical risk $R_{emp}(\alpha)$ and a confidence interval $CI(\alpha)$ containing the Vapnik-Chervonenkiss dimension of the function space $S$.

This dependency is shown on Fig. 2.

Instead of optimizing $\alpha$ over an arbitrary set of possible parameters $S$, we use the bounds to find a set $S^*$ for which the risk's bound is minimal, and then perform the search for the solution $\alpha^*$ within $S^*$. For more details, see [61].

The hierarchical learning approach, by reducing the complexity of the original learning problem by decomposing it into simpler ones, tries to optimize the corresponding search spaces on subsequent levels of the learning hierarchy, and is analogous in function to the SRM technique. One can consider decomposition as one of possible strategies in SRM aimed at searching for (sub)optimal spaces. The resulting space corresponds to the family of searching spaces obtained on different levels of decomposition. For any submodel on the $i + 1$-th level the searching space for solutions is discovered on the basis of some search spaces from the $i$-th level. The search for (sub)optimal decomposition is conducted by minimization of the description length of solutions from spaces on different decomposition levels while preserving the satisfactory quality of solutions. The searching spaces for approximation of concepts from any level $i+1$ on the basis of concepts from the level $i$ of decomposition are computationally feasible because any two successive levels of decomposition should be, in a sense, semantically *close* [30]. This means that the searching spaces brought about on particular

[1] Searching for any $f_i$ (not corresponding to the leaf decomposition level) is performed over the space constructed on the basis of some already discovered spaces linked to some submodels from the predecessor decomposition level relative to $f_i$.

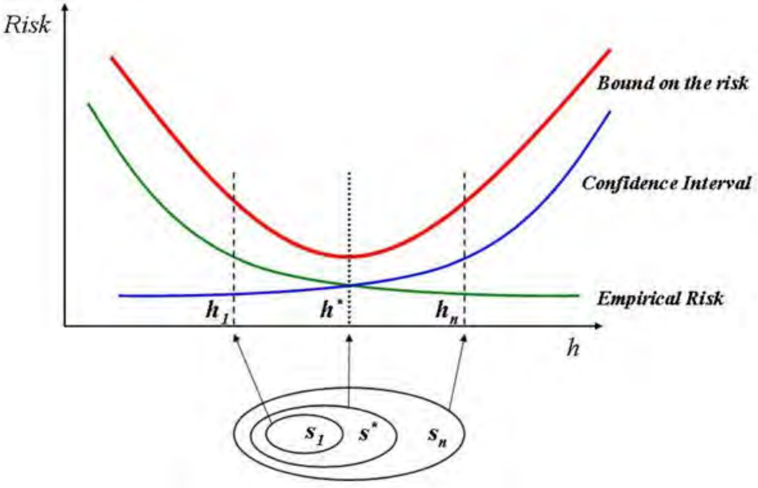


**Fig. 2.** Actual risk bounds across search spaces.(Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1999)

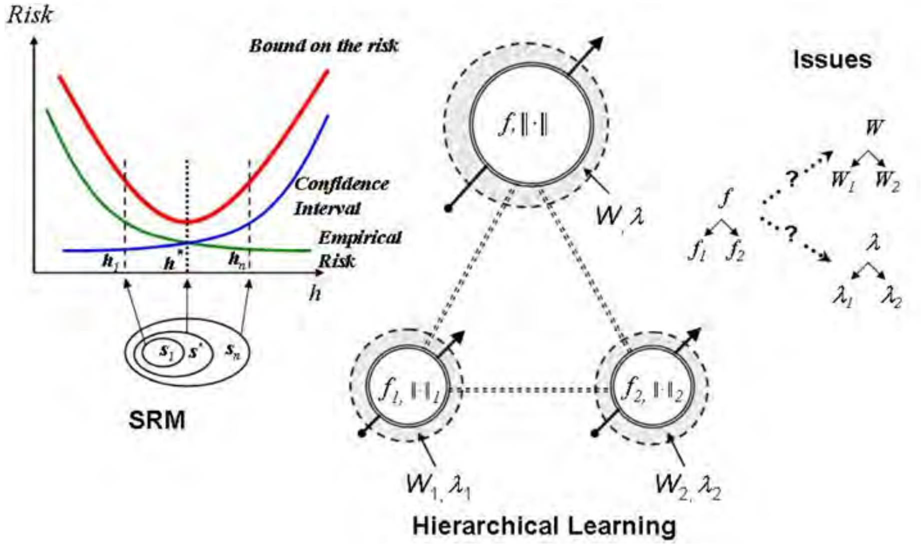


**Fig. 3.** SRM vs Hierarchical Learning

decomposition levels are smaller than those obtained without decomposition. Moreover SRM can be used on each particular decomposition level to optimize the searching space for approximation of concepts on this level. For details, see Fig. 3.

Another advantage of the hierarchical learning model lies in the construction of the descriptive language in which classifiers on subsequent levels are built. The choice of language directly influences the potential search space and is therefore crucial for classifier building. With a proper reasoning scheme in place, we can construct the descriptive language on a higher level from those already established on lower levels, which proves effective in reducing the learning time and boosting the overall learning performance. The choice of language can be interpreted as a step in searching for sub(optimal) spaces in SRM.

### 2.3 Ontology Matching

The knowledge on training samples that comes from an expert obviously reflects his perception about the samples. The language used to describe this knowledge is a component of the expert's ontology which is an integral part of his perception. In a broad view, an ontology consists of a vocabulary, a set of concepts organized in some kind of structures, and a set of binding relations amongst those concepts [15]. We assume that the expert's ontology when reasoning about complex structured samples will have the form of a multi-layered hierarchy, or a *lattice*, of concepts. A concept on a higher level will be synthesized from its children concepts and their binding relations. The reasoning thus proceeds from the most primitive notions at the lowest levels and work bottom-up towards more complex concepts at higher levels.

Hierarchical learning, together with the transfer of knowledge expressed in natural languages from external experts to low-level computer operators, constitutes an excellent illustration of *Granular Computing* in action.

### 2.4 External Knowledge Transfer

The knowledge elicitation process assumes that samples, for which the learning system deems it needs additional explanations, are submitted to the expert, which returns not only their correct class identity, but also an explanation on *why*, and perhaps more importantly, *how* he arrived at his decision. This explanation is passed in the form of a rule:

$$[CLASS(u) = k] \equiv \Im(EFeature_1(u), ..., EFeature_n(u)),$$

where $EFeature_i$ represents the expert's perception of some characteristics of the sample $u$, while synthesis operator $\Im$ represents his perception of some relations between these characteristics. In a broader view, $\Im$ constitutes of a *relational structure* that encompasses the hierarchy of experts' concepts expressed by $EFeature_i$.

The ontology matching aims to translate the components of the expert's ontology, such as $EFeature_i$ and binding relations embedded in the $\Im$ structure, expressed in the foreign language $L_f$, into the patterns (or classifiers) expressed in a language familiar to the learning system, e.g:

- $[FaceType(Ed) = Square] \equiv$(Ed.Face().Width - Ed.Face().Height) $\leq$ 2cm,
- $[Eclipse(p) = True] \equiv$ (s=p.Sun())$\wedge$(m=p.Moon())$\wedge$(s$\cap$m.Area$\geq$ s.Area$\cdot$0.6).

Here the abstract concepts such as “Ed has a square face” or “The Sun is in eclipse” get translated into classification rules built from computable measurements and observation features.

As the human perception is inherently prone to variation and deviation, the concepts and relations in a human expert's ontology are approximate by design. To use the terms of granular computing, they are information granules that encapsulate the autonomous yet interdependent aspects of human perception.

The matching process, while seeking to accommodate various degrees of variation and tolerance in approximating those concepts and relations, will follow the same hierarchical structure of the expert's reasoning. This allows parent concepts to be approximated using the approximations of children concepts, essentially building a *layered approximate reasoning scheme*. Its hierarchical structure provides a natural realization of the concept of granularity, where nodes represent clusters of samples/classifiers that are similar within a degree of resemblance/functionality, while layers form different levels of abstraction/perspectives on selected aspects of the sample domain.

On the other hand, with such an established multi-layered reasoning architecture, we can take advantages of the results obtained within the Granular Computing paradigm, which provides frameworks and tools for the fusion and analysis of compound information granules from previously established ones, in a straightforward manner. The intermediate concepts used by external experts to explain their perception are vague and ambiguous, which makes them natural subjects to granular calculi.

The translation must

- allow for a flexible matching of a variations of similar domestic patterns to a foreign concept, i.e. the translation result should not be a single patterns, but rather a collection or cluster of patterns.
- find approximations for the foreign concepts and relations, while preserving their hierarchical structure. In other words, inherent structure of the provided knowledge should be intact.
- ensure robustness, which means independence from noisy input data and incidental underperformance of approximation on lower levels, and stability, which guarantees that any input pattern matching concepts on a lower level to a satisfactory degree will result in a satisfactory target pattern on the next level.

We assume an architecture that allows a learning system to consult a human expert for advices on how to analyze a particular sample or a set of samples. Typically this is done in an iterative process, with the system subsequently incorporating knowledge elicited on samples that could not be properly classified in previous attempts [32]. (See Fig. 4 below).

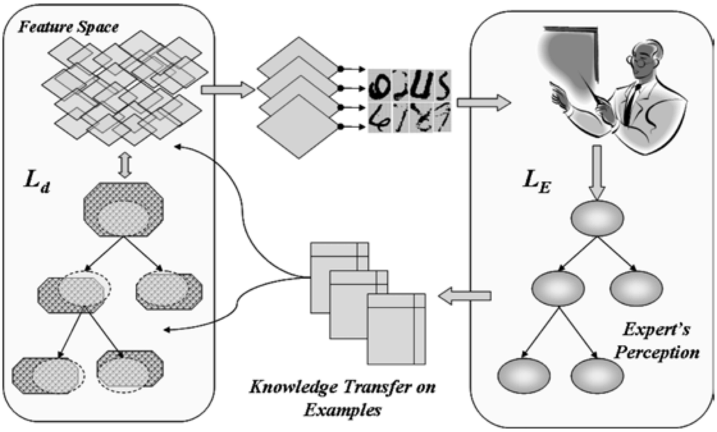


**Fig. 4.** Expert's knowledge elicitation

### 2.4.1 Approximation of Concepts

A foreign concept $C$ is approximated by a domestic pattern (or a set of patterns) $p$ in term of a rough inclusion measure $Match(p, C) \in [0, 1]$. Such measures take root in the theory of rough mereology [45], and are designed to deal with the notion of inclusion to a degree. An example of concept inclusion measures would be:

$$Match(p, C) = \frac{|\{u \in T : Found(p, u) \wedge Fit(C, u)\}|}{|\{u \in T : Fit(C, u)\}|},$$

where $T$ is a common set of samples used by both the system and the expert to communicate with each other on the nature of expert's concepts, $Found(p, u)$ means a pattern $p$ is present in $u$ and $Fit(C, u)$ means $u$ is regarded by the expert as fit to his concept $C$.

Our principal goal is, for each expert's explanation, find sets of patterns $Pat$, $Pat_1$,...,$Pat_n$ and a relation $\Im_d$ so as to satisfy the following *quality requirement*:

$$\textbf{if } (\forall i : Match(Pat_i, EFeature_i) \geq p_i) \wedge (Pat = \Im_d(Pat_1, ..., Pat_n))$$
$$\textbf{then } Quality(Pat) > \alpha,$$

where $p, p_i : i \in \{1, .., n\}$ and $\alpha$ are certain cutoff thresholds, while the $Quality$ measure, intended to verify if the target pattern $Pat$ fits into the expert's concept of sample class $k$, can be any, or combination, of popular quality criteria such as *support*, *coverage*, or *confidence* [46], where

$$Support_{CLASS=k}(Pat) = |\{u \in U : Found(Pat, u) \wedge CLASS(u) = k\}|,$$

$$Confidence_{CLASS=k}(Pat) = \frac{Support(Pat)}{|\{u \in U : Found(Pat, u)\}|},$$

$$Coverage_{CLASS=k}(Pat) = \frac{Support(Pat)}{|\{u \in U : CLASS(u) = k\}|},$$

and $U$ is the training set.

In other words, we seek to translate the expert's knowledge into the domestic language so that to generalize the expert's reasoning to the largest possible number of training samples. More refined versions of the inclusion measures would involve additional coefficients attached to e.g. $Found$ and $Fit$ test function. Adjustment of these coefficients based on feedback from actual data may help optimize the approximation quality.

For example, let's consider a handwritten digit recognition task:

When explaining his perception of a particular digit image sample, the expert may employ concepts such as *Circle*, *Vertical Strokes* or *West Open Belly.* The expert will explain what he means when he says, e.g. $Circle$, by providing a decision table $(U, d)$ with reference samples, where $d$ is the expert decision to which degree he considers that $Circle$ appears in samples $u \in U$. The samples in $U$ may be provided by the expert, or may be picked up by him among samples explicitly submitted by the system, e.g. those that had been misclassified in previous attempts.

The use of rough inclusion measures allows for a very flexible approximation of foreign concept. A stroke at 85 degree to the horizontal in a sample image can still be regarded as a vertical stroke, though obviously not a 'pure' one. Instead of just answering in a $Yes/No$ fashion, the expert may express his degrees of belief using such natural language terms as $Strong$, $Fair$, or $Weak$ (See Fig. 5).

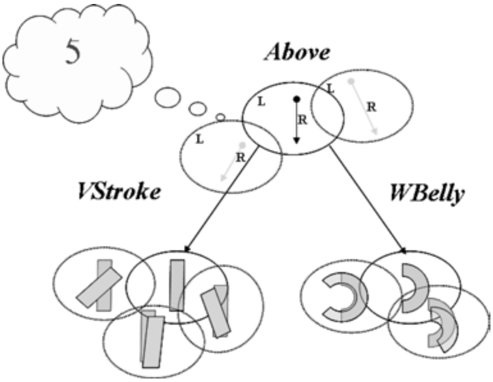


**Fig. 5.** Tolerant matching by expert

The expert's feedback will come in the form of a decision table (See Table 1).

**Table 1.** Perceived features

| | $Circle$ |
|---|---|
| $u_1$ | $Strong$ |
| $u_2$ | $Weak$ |
| ... | ... |
| $u_n$ | $Fair$ |

**Table 2.** Translated features

| | $DPat$ | $Circle$ |
|---|---|---|
| $u_1$ | 252 | $Strong$ |
| $u_2$ | 4 | $Weak$ |
| ... | ... | ... |
| $u_n$ | 90 | $Fair$ |

The translation process attempts to find domestic feature(s)/pattern(s) that approximate these degrees of belief (e.g. such as presented in Table 2). Domestic patterns satisfying the defined quality requirement can be quickly found, taking into account that sample tables submitted to experts are usually not very large. Since this is essentially a rather simple learning task that involves feature selection, many strategies can be employed. In [34], genetic algorithms equipped with some greedy heuristics are reported successful for a similar problem. Neural networks also prove suitable for effective implementation.

It can be observed that the intermediate concepts like *Circle* or *Vertical Strokes*, provided by a human expert, along with satisfiability assessments like $Strong$, $Fair$, or $Weak$ form information granules within the perception of the expert. The granules correspond to different levels of abstraction, or focus, of his reasoning about a particular class of samples. The translation process transforms these information granules into classifiers capable of matching particular parts of actual samples with intermediate expert's concepts, which essentially incorporates the human perception, by way of using information granules, into the learning process.

### 2.4.2 Approximation of Relations

The approximation of higher level relations between concepts has been formalized within the framework of perception structures, recently developed in [49]. A *perception structure* $S$, in a simpler form, is defined as:

$$S = (U, M, F, \models, p),$$

where U is a set of samples, $F$ is a family of formulas expressed in domestic language that describe certain features of the samples and $M$ is a family of relational structures in which these formulas can be evaluated, while $p\colon U \rightarrow M \times F$ is a *perception function* such that $\forall u{\in}U : p_1(u){\models}p_2(u)$ ($p_1$ and $p_2$ are the first and second component projections of $p$) which means that $p_2(u)$ is satisfied (is true) in the relational structure $p_1(u)$. This may express that some relations among features within samples are observed.

For a given sample $u$, we define a set

$$M(u) = \{R{\in}M : R \models p_2(u)\},$$

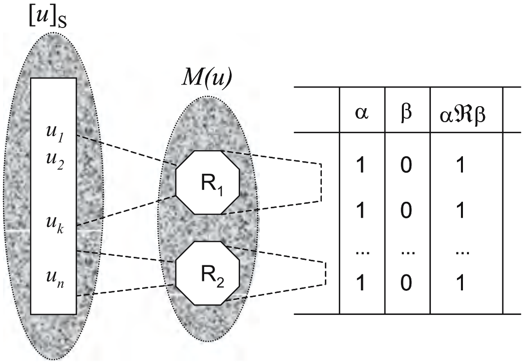
which contains all possible relational structures for which formulas, or in other words, features observed in $u$ yield.

### 2.4.3 Approximate Clusters

Given a perception structure $S$, an approximate cluster of a given sample $u$ is defined as:

$$[u]_S = \bigcup_{R \in M(u)} p_1^{-1}(R).$$

This cluster contains samples from $U$ that have similar structures to $u$, with regard to the perception $p$, i.e. those with similar relational structures that also hold true the features observed in $u$ (See Fig. 6).



| α | β | $\alpha\Re\beta$ |
|---|---|---|
| 1 | 0 | 1 |
| 1 | 0 | 1 |
| ... | ... | ... |
| 1 | 0 | 1 |

**Fig. 6.** Approximate cluster

For example, if we construct a perception structure that contains a formula describing a part of a digit is *above* another part, then within this perception, the approximate cluster of a digit '6', which has a slant stroke over a circle, would comprise of all digits that have similar structure, i.e. containing a slant stroke over a circle.

Perception structures, following natural constructs in the expert's foreign language, should involve tolerant matching. Let's suppose that we allow a $soft$ perception on samples of $U$ by introducing a similarity relation $\tau$ between them. This relation, for example might assume that two samples resemble each other to a degree. This naturally leads to clusters of similar relational structures in $M$. With samples now perceived as similar to each other in a degree, we shall allow for a similarity relation in $M$. Two relational structures might be considered approximately *the same* if they allow for similar formulas to yield similar results in majority of cases when these formulas are applicable. The family $M$ thus becomes granulated by $\tau$ and is denoted by $M_\tau$.

The same follows for the family $F$ of features, or formulas that, for instance, do not always have the same value, but are equivalent in most cases, or in all or majority of a cluster of similar relational structures. Formulas' evaluation might be extended to comprise degrees of truth values, rather than plain binary constants. The family $F$ hence becomes granulated with regards to $\tau$ and is denoted by $F_\tau$.

The perception structure S hence becomes, for a given similarity measure $\tau$ in $U$: $S = (U, M_\tau, F_\tau, \models, p)$ which permits a much more flexible space and a variety of methods for concept approximation.

In the above mentioned example, a similarity induced perception might consider as the approximate cluster of a digit '5' the set of every samples that will have a *stroke* over a *closed curve* (not just slant strokes and circles as before). Moreover, the new perception also allows for a greater variation of configurations considered to fit into the concept of *above*.

The definition of an approximate cluster becomes:

$$[u]_S = \bigcup_{R \in M_\tau(u)} p_1^{-1}(R).$$

The task of approximating an expert's concept involving relations between components is now equivalent to finding a perception function that satisfies some quality criteria. Let's suppose that the expert provide us a set $C$ of samples he considers fit to his concept. We have to find a perception function $p$ such that:

$$Confidence : \frac{|[u]_S \cap C|}{|[u]_S|} > c,$$

and/or

$$Support : \frac{|[u]_S \cap C|}{|U|} > s,$$

where $u$ is some sample from $C$, and $0 < c, s < 1$.

Having approximated the expert's features $EFeature_i$, we can try to translate his relation $\Im$ into our $\Im_d$ by asking the expert to go through $U$ and provide us with the additional attributes of how strongly he considers the presence of $EFeature_i$ and to what degree he believes the relation $\Im$ holds. Again, lets consider the handwritten recognition case.(See Table 3).

**Table 3.** Perceived relations

| | *VStroke* | *WBelly* | *Above* |
|---|---|---|---|
| $u_1$ | *Strong* | *Strong* | *Strong* |
| $u_2$ | *Fair* | *Weak* | *Weak* |
| ... | ... | ... | ... |
| $u_n$ | *Fair* | *Fair* | *Weak* |

**Table 4.** Translated relations

| | #V_S | #NES | $S_y < B_y$ | *Above* |
|---|---|---|---|---|
| $u_1$ | 0.8 | 0.9 | (*Strong*,1.0) | (*Strong*, 0.9) |
| $u_2$ | 0.9 | 1.0 | (*Weak*, 0.1) | (*Weak*, 0.1) |
| ... | ... | ... | ... | ... |
| $u_n$ | 0.9 | 0.6 | (*Fair*, 0.3) | (*Weak*, 0.2) |

We then replace the attributes corresponding to $EFeature_i$ with the rough inclusion measures of the domestic feature sets that approximate those concepts (computed in the previous step). In the next stage, we try to add other features, possibly induced from original domestic primitives, in order to approximate the decision $d$. Such a feature may be expressed by $S_y < B_y$, which tells whether the median center of the stroke is placed closer to the upper edge of the image than the median center of the belly. (See Table 4).

The expert's perception *A '6' is something that has a 'vertical stroke' 'above' a 'belly open to the west'* is eventually approximated by a classifier in the form of a rule:

**if** $S$(#BL_SL $> 23$) **AND** $B$(#NESW $> 12\%$) **AND** $S_y < B_y$ **then** CL='6',

where $S$ and $B$ are designations of pixel collections, #BL_SL and #NESW are numbers of pixels with particular topological feature codes, and $S_y < B_y$ reasons about centers of gravity of the two collections.

Approximate reasoning schemes embody the concept of information granularity by introducing a hierarchical structure of abstraction levels for the external knowledge that come in the form of a human expert's perception. The granularity helps to reduce the cost of the knowledge transfer process, taking advantage of the expert's hints. At the same time, the hierarchical structure ensures to preserve approximation quality criteria that would be hard to obtain in a flat, single-level learning process.

From yet another perspective, the reasoning schemes that encompass a human expert's intermediate concepts like *Vertical Stroke*, *Above* and their satisfability assessments such as *Strong* or *Fair* represents the way humans reason about samples through different levels of abstraction. The connections between intermediate concepts and transitions from lower to upper levels allow to shift the perception focus from smaller parts of objects to more abstract, global features. These reasoning schemes also provide off-the-shelf recipes as to how to assemble more compound information granules from simpler, already established ones. Translated into domestic languages, they become powerful classifiers that help expand the human perception structures to actual samples.

### 2.5 Outliers

Conceptually, *outliers/exceptions* are kind of atypical samples that stand out from the rest of their group or behave very differently from the norm [1]. While there is still no universally accepted formal definition of being an outlier, several descriptions seem to reflect the essential spirit. According to Hawkin: *An outlier is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism*, while Barnett and Lewis define an outlier as *an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.* [3]. These samples previously would usually be treated as bias or noisy input data and were frequently discarded or suppressed in subsequent analyses. However, the rapid development of Data Mining, which aims to extract from data as much knowledge as possible, has made outlier identification and analysis one of its principal branches. Dealing with outliers is crucial to many important fields in real life such as fraud detection in electronic commerce, intrusion detection, network management, or even space exploration. At the same time, there is an increasing effort in the Machine Learning community to develop better methods for outlier detection/analysis, as outliers often carry useful subtle hints on the characteristics of the sample domain and, if properly analyzed, may provide valuable guidance in discovering the causalities underlying the behavior of a learning system. As such, they may prove valuable as an additional source of search control knowledge and as a mean for the construction of better classifiers.

Most popular measures to detect outliers [19] are based on either probabilistic density analysis [12] or distance evaluation [28]. Knorr made an attempt to elicit intensional knowledge from outliers through the analysis of the dynamicity of outliers' set against changes in attribute subsets [27]. However, no thorough model or scheme for the discovery of intensional knowledge from identified outliers has been established. In particular, there is almost no known attempt to develop methods for outlier analysis amongst structured objects, i.e. objects that display strong inner dependencies between theirs own features or components. Perhaps the reason for this is the fact that while many elaborated computation models for the detection of outliers have been proposed , their effective use in eliciting additional domain knowledge, as well as the elicitation of intensional knowledge within outliers, is believed difficult without support of a human expert.

In this paper, we approach the detection and analysis of outliers in data from a Machine Learning perspective. We propose a framework based on the Granular Computing paradigm, using tools and methods originated from Rough Set and Rough Mereology theories. The process of outlier detection is refined by the evaluation of classifiers constructed employing intensional knowledge elicited from suspicious samples. The internal structures of the sample domain will be dealt with using hierarchical approximate reasoning schemes and layered learning. We show the role of an external domain knowledge source by human experts in outlier analysis, and present methods for the successful assimilation of such knowledge. Introduced methods and schemes are illustrated with an example handwritten digit recognition system.

Most existing outlier identification methods employ either probabilistic density analysis, or distance measures evaluation [19]. Probabilistic approach typically run a series of statistical discordancy tests on a sample to determine whether it can be qualified as an outlier. Sometimes this procedure is enhanced by a dynamic learning process. Their main weakness is the assumption of an underlying distribution of samples, which is not always available in many real life applications. Difficulties with their scalability in numbers of samples and dimensions are also a setback of primary concern.

Another approach to outlier detection relies on certain distance measures established between samples. Known methods are data clustering and neighbor analysis. While this approach can be applied to data without any assumed a priori distribution, they usually entails significant computation costs.

Let $C_k$ be a cluster of samples for class $k$ during the training phase and $d_k$ be the distance function established for that class. For a given cut-off coefficient $\alpha \in (0, 1]$, a sample $u^*$ of class $k$ is considered "difficult", "hard" or "outlier" if, e.g:

$$d_k(u^*, C_k) \geq \alpha \cdot \max\{(v, C_K) : v \in TR \wedge CLASS(v) = k\},$$

which means $u^*$ is far from the "norm" in term of its distance to the cluster center, or

$$|\{v : v \in C_k \wedge d_k(u^*, v) \leq d_k(v, C_k)\}| \leq \alpha \cdot |C_k|,$$

which means $u^*$ is amongst the most outreaching samples of the cluster.

Another popular definition of outlier is:

$$|\{v : v \in C_k \wedge d_k(u^*, v) \geq D\}| \leq \alpha \cdot |C_k|,$$

which means at least a fraction $\alpha$ of objects in $C_k$ lies in a greater distance than $D$ from $u^*$.

It can be observed that both approaches pay little attention to the problem of eliciting intensional knowledge from outliers, meaning no elaborated information that may help explain the reasons why a sample is considered outlier. This kind of knowledge is important for the validity evaluation of identified outliers, and certainly is useful in improving the overall understanding of the data.

Knorr and Ng made an attempt to address this issue by introducing the notion strength of outliers, derived from an analysis of dynamicity of outlier sets against changes in the features' subsets [26, 27]. Such analyzes belong to the very well established application domain of Rough Sets, and indeed a formalization of a similar approach within the framework of Rough Sets has been proposed by [23].

Our approach to outlier detection and analysis will assume a somewhat different perspective. It focuses on two main issues:

1. Elicitation of intensional knowledge from outliers by approximating the perception of external human experts.
2. Evaluation of suspicious samples by verification the performance of classifiers constructed using knowledge elicited from these samples.

Having established a mechanism for eliciting expert's knowledge as described in previous sections, we can develop outlier detection tests that might be completely independent from the existing similarity measures within the learning system as outlined in the Fig. 7 below:

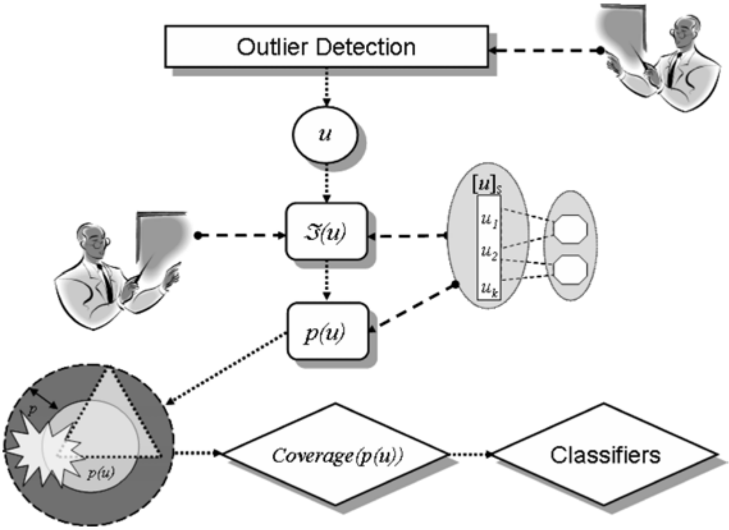


**Fig. 7.** Outlier analysis scheme

For a given training sample $u^*$,

**Step 1.** We ask the expert for his explanation on $u^*$. **Step 2.** The expert provides a foreign knowledge structure $\Im(u^*)$. **Step 3.** We approximate $\Im(u^*)$ under restrictive matching degrees to ensure only the immediate neighborhood of $u^*$ is investigated. Let's say the result of such an approximation is a pattern (or set of patterns) $p_u^*$. **Step 4.** It is now sufficient to check $Coverage(p_u^*)$. If this coverage is high, it signifies that $u^*$ may bear significant information that is also found in many other samples. The sample $u^*$ therefore cannot be regarded as an outlier despite the fact that there may not be many other samples in its vicinity in terms of existing domestic distance measures of the learning system.

This test shows that distance-based outlier analysis and expert's elicited knowledge are complementary to each other.

In our architecture, outliers may be detected as samples that defied previous classification efforts, or samples that pass the above described outlier test, but may also be selected by the expert himself. This helps the classification system to focus on difficult samples in order to gradually improve the overall performance, in a way similar to that of popular boosting or leveraging algorithms. The main difference is that boosting algorithms employ a priori formulas/strategies to adjust weights to positive and negative samples, whereas our approach relies on the domain knowledge elicited from the external expert. In this way, we can benefit from the best of both sources of knowledge.

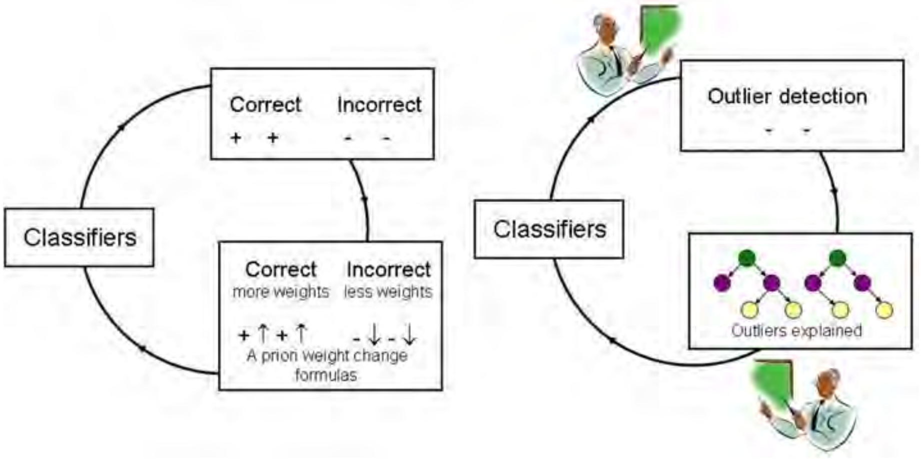


**Fig. 8.** Boosting vs Hierarchical Learning

### 2.6 Reinforcement Learning and Planning

The next two examples illustrate approximation of compound concepts in reinforcement learning and planning.

In reinforcement learning [13, 24, 29, 40, 43, 53, 58], the main task is to learn the approximation of the function $Q(s, a)$, where $s, a$ denotes a global state of the system and an action performed by an agent $ag$ and, respectively and the real value of $Q(s, a)$ describes the reward for executing the action $a$ in the state $s$. To approximate the function $Q(s, a)$, probabilistic models are used. However, for compound real-life problems it may be hard to build such models for such a compound concept as $Q(s, a)$ [60]. We propose another approach to the approximation of $Q(s, a)$ based on ontology approximation. The approach is based on the assumption that in a dialog with experts additional knowledge can be acquired, making it possible to create a ranking of values $Q(s, a)$ for different actions $a$ in a given state $s$. The explanation given by expert about possible values of $Q(s, a)$ may involve concepts from a special ontology. Using this ontology one can follow hierarchical learning methods to learn the approximations of its concepts. Such concepts can have a temporal character as well. This means the ranking of actions may depend not only on the current action and state but also on actions performed in the past, as well as on the changes caused by these actions.

In [6, 7] a computer tool based on rough sets for supporting automated planning of the medical treatment (see, e.g., [18, 59]) is discussed. In this approach, a given patient is treated as an investigated complex dynamical system, whilst diseases of this patient (RDS, PDA, sepsis, Ureaplasma and respiratory failure) are treated as compound objects changing and interacting over time. As a measure of planning success (or failure) in the experiments, we use a special hierarchical classifier that can predict the similarity between two plans as a number between 0.0 and 1.0. This classifier has been constructed on the basis of the special ontology specified by human experts and data sets. It is important to mention that in addition to the ontology, experts also provided the exemplary data (values of attributes) for the purpose of concepts approximation. The methods of construction such classifiers are based on approximate reasoning schemes (AR schemes, for short) and were described, e.g., in [5, 8, 9, 31]. We applied this method for approximation of similarity between plans generated in automated planning and plans proposed by human experts during the realistic clinical treatment.

### 2.7 Interaction with the Web

Let us discuss shortly problems which can be solved by human in dialog with the Web. Examples of such problems are considered in Service Oriented Computing or Service Oriented Architecture (see, e.g., [16]). Assuming that this dialog is performed in a simplified fragment of natural language [62, 64] one should develop tools for approximation concepts used in dialog to make them available to the Web for approximate reasoning in searching for the solutions of problems. One can expect that in a near future the Web can automatically help the users to synthesize required services if it will be possible to understood to satisfactory degree the specification received in dialog with users.

## 3 Selected Advanced Issues on GC

In this section, we discuss some advanced issues on GC. They are related to granules represented by agents or teams of agents interacting in changing environments.

We start from a general discussion on the *Wisdom technology* (wistech) system outlined recently in [21, 22].

Wisdom commonly means *rightly judging* based on available knowledge and interactions. This common notion can be refined. By *wisdom*, we understand an adaptive ability to make judgments correctly (in particular, correct decisions) to a satisfactory degree, having in mind real-life constraints. The intuitive nature of wisdom understood in this way can be metaphorically expressed by the so-called *wisdom equation* as shown in (4).

$$wisdom = adaptive\ judgment\ +\ knowledge\ +\ interaction. \tag{4}$$

Wisdom can be treated as a certain type of knowledge. In particular, this type of knowledge is important at the highest level of hierarchy of meta-reasoning in intelligent agents.

Wistech is a collection of techniques aimed at the further advancement of technologies to acquire, represent, store, process, discover, communicate, and learn *wisdom* in designing and implementing intelligent systems. These techniques include approximate reasoning by agents or teams of agents about vague concepts concerning real-life, dynamically changing, usually distributed systems in which these agents are operating. Such systems consist of other autonomous agents operating in highly unpredictable environments and interacting with each others. Wistech can be treated as the successor of database technology, information management, and knowledge engineering technologies. Wistech is the combination of the technologies represented in equation (4) and offers an intuitive starting point for a variety of approaches to designing and implementing computational models for wistech in intelligent systems.

- *Knowledge technology* in wistech is based on techniques for reasoning about knowledge, information, and data, techniques that enable to employ the current knowledge in problem solving. This includes, e.g., extracting relevant fragments of knowledge from knowledge networks for making decisions or reasoning by analogy.
- *Judgment technology* in wistech is covering the representation of agent perception and adaptive judgment strategies based on results of perception of real life scenes in environments and their representations in the agent mind. The role of judgment is crucial, e.g., in adaptive planning relative to the Maslow Hierarchy of agents' needs or goals. Judgment also includes techniques used for perception, learning, analysis of perceived facts, and adaptive refinement of approximations of vague complex concepts (from different levels of concept hierarchies in real-life problem solving) applied in modeling interactions in dynamically changing environments (in which cooperating, communicating, and competing agents exist) under uncertain and insufficient knowledge or resources.
- *Interaction technology* includes techniques for performing and monitoring actions by agents and environments. Techniques for planning and controlling actions are derived from a combination of judgment technology and interaction technology.

The wistech system is strongly related to the idea of Gottfried Wilhelm Leibniz, one of the greatest mathematicians. He has discussed, in a sense, calculi of thoughts. In particular, he has written

> *If controversies were to arise, there would be no*
> *more need of disputation between two philosophers*
> *than between two accountants. For it would suffice to*
> *take their pencils in their hands, and say to each other:*
> *'Let us calculate'.*
> – Gottfried Wilhelm Leibniz,
> Dissertio de Arte Combinatoria (Leipzig, 1666).

> *... Languages are the best mirror of the human mind,*
> *and that a precise analysis of the signification of words*
> *would tell us more than anything else about the operations*
> *of the understanding.*
> – Gottfried Wilhelm Leibniz,
> New Essays on Human Understanding (1705)
> Translated and edited by
> Peter Remnant and Jonathan Bennett
> Cambridge: Cambridge UP, 1982

Only much later, it was possible to recognize that new tools are necessary for developing such calculi, e.g., due to the necessity of reasoning under uncertainty about objects and (vague) concepts. Fuzzy set theory (Lotfi A. Zadeh, 1965) and rough set theory (Zdzisław Pawlak, 1982) represent two complementary approaches to vagueness. Fuzzy set theory addresses gradualness of knowledge, expressed by the fuzzy membership, whereas rough set theory addresses granularity of knowledge, expressed by the indiscernibility relation. Granular computing (Zadeh, 1973, 1998) may be now regarded as a unified framework for theories, methodologies and techniques for modeling of calculi of thoughts based on objects called granules.

There are many ways to build foundations for wistech computational models. One of them is based on the *rough-granular computing* (RGC) [52]. Rough-granular computing (RGC) is an approach for constructive definition of computations over objects called granules, aiming at searching for solutions of problems which are specified using vague concepts. Granules are obtained in a process called granulation. Granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem-solving [64]. The proposed approach combines rough set methods with other soft computing methods, and methods based on granular computing (GC). RGC is used for developing one of the possible wistech foundations based on approximate reasoning about vague concepts.

Let us discuss some issues important to wistech, pertaining to compound granules which can perceive, and interact with, their environments.

### 3.1 Compound Granules in Perception and Interaction

Perception and interaction are closely related issues. Let us assume that an agent $ag$ is perceiving the environment state $e$. The results of perceiving (e.g., by sensors) of different parts of $e$ are stored in a a generalized information system $IS$ (see Sect. 1.2). For each such a part $s$ a partial information $Inf(s)$ is stored in $IS$ together with information on relationships between parts. Form $IS$ the structural model $M_e$ of $e$ is derived by hierarchical modeling. Granule $M_e$ can be represented by a relational structure or a cluster of such structures). Next, the structural model $M_e$ is matched against knowledge base network of the agent $ag$. The result of matching is a family $\mathcal{F}_e$ of concepts together with information about degrees to which these concepts are satisfied. Now, the judgment engine of $ag$ is used to predict the current goals and to select of the relevant action (plan or communication) for required for interaction with the environment. Agent $ag$ is attempting to make necessary changes or to move in the environment to reach the target goal. Note the environment changes are reflected by changes in the internal state representation of $ag$. Moreover, information $Inf(s')$, $M'_s$ or $\mathcal{F}_{s'}$ about the next state $e'$ of the environment is predicted, where $s'$ denotes a part of $e'$ perceived by $ag$. This information is further compared with the perceived information about the next state $e'$, e.g., for reconstructing the current plan. The judgment engine of $ag$ is also used for predicting changes of the internal state of $ag$ caused by the actual environment state and by the actual internal state [30].

In [30] the above discussed judgment process was disussed in the framework of approximate reasoning about changes of the environment state and the agent state. In particular, this calls for approximation of complex functions characterizing these changes.

### 3.2 Coalition Formation and Interactive Computations

Coalitions of granules play an important role in interactive computations and reasoning about such computations. One can consider coalitions as operations on collections of granules representing agents to granules representing a meta-agent, i.e., a team of agents. Any such operation should provide for the construction of (i) the perception mechanism of coalition from the perception mechanisms of its members; (ii) the coalition judgment engine using the judgment engines of the coalition members; (iii) the coalition knowledge base network from knowledge base networks of the coalition members; (iv) the interaction mechanism of the coalition from the interaction mechanisms of the coalition members. For example, in a given situation, each member of coalition can have several choices for actions but the finally selected action by each member is based on cooperation of coalition members, i.e., they are choosing actions on the basis of, e.g., protocol of cooperation. Moreover, the environment is perceiving coalition as a whole, e.g., any action performed by coalition can be treated as the results of a vector $(a_1, \ldots, a_n)$ of actions of its members. These actions are selected on the basis of the accepted protocol. Note also perceived features of coalition refer to the coalition as the whole rather than its parts.

There are several challenging issues related to coalitions such as learning strategies for (hierarchical) coalition formation, constructing language of actions, plans, communication for coalition on the basis of languages of its members. Among these issues hierarchical coalitions play a special role in approximate reasoning about interactive computations, making it possible to perform approximate reasoning about the interaction of the whole system with the environment.

### 3.3 Granule Representation and Adaptation of Knowledge Base Networks

In the matching process of structural models with knowledge base networks, an important role play strategies of granule representations and geometry of granules [17].

Observe that during the perception, matching, and judgment processes the current knowledge base network may be updated. For example, some new patterns, concepts, production rules, rules, or approximate reasoning schemes are discovered and stored. Some other patterns may be removed from the knowledge base networks [48, 65].

### 3.4 Scalability Issues and Language Evolution

One can formulate several important questions related to languages used for communication by agents. Here are some examples. How the names for concepts are created? Why they are created? When we should go beyond ontologies of concepts? How the communication languages of agents are evolving when agents are dealing with the scalability issues, i.e., when the agents are trying to move from solving small size problems to large size problems?

## 4 Conclusions

We discussed the role of GC in approximation of complex concepts and in interactive computations, e.g., performed by agents interacting with environments and among themselves. We emphasized the role of dialogs of human experts with different granular systems in improving their performance. Finally, we formulated several challenges for approximate reasoning in granular systems interacting with the environments, in particular with other agents or human experts.

**Acknowledgments.** The research has been partially supported by the grant N N516 368334 from Ministry of Science and Higher Education of the Republic of Poland and by the grant “Decision support – new generation systems” of Innovative Economy Operational Programme 2008-2012 (Priority Axis 1. Research and development of new technologies) managed by Ministry of Regional Development of the Republic of Poland.

## References

1. Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. In: Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, pp. 37–46. ACM Press, New York (2001)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
3. Barnett, V., Lewis, T.: Outliers in Statistical Data. John Wiley and Sons Ltd, Chichester (1978)
4. Barwise, J., Seligman, J.: Information Flow: The Logic of Distributed Systems. Cambridge University Press, Cambridge (1997)
5. Bazan, J.: Rough sets and granular computing in behavioral pattern identification and planning. In: Pedrycz, et al. (eds.) [39], pp. 777–800.
6. Bazan, J., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Automatic planning of treatment of infants with respiratory failure through rough set modeling. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) RSCTC 2006. LNCS, vol. 4259, pp. 418–427. Springer, Heidelberg (2006)
7. Bazan, J., Kruczek, P., Bazan-Socha, S., Skowron, A., Pietrzyk, J.J.: Risk pattern identification in the treatment of infants with respiratory failure through rough set modeling. In: Proceedings of Information Processing and Management under Uncertainty in Knowledge-Based Systems (IPMU 2006), Paris, France, July 2-7, 2006, vol. (3), pp. 2650–2657. Editions E.D.K. (2006)
8. Bazan, J., Peters, J.F., Skowron, A.: Behavioral pattern identification through rough set modelling. In: Ślęzak, D., Yao, J., Peters, J.F., Ziarko, W.P., Hu, X. (eds.) RSFDGrC 2005. LNCS, vol. 3642, pp. 688–697. Springer, Heidelberg (2005)
9. Bazan, J., Skowron, A.: Classifiers based on approximate reasoning schemes. In: Dunin-Kęplicz, B., Jankowski, A., Skowron, A., Szczuka, M. (eds.) Monitoring, Security, and Rescue Techniques in Multiagent Systems. Advances in Soft Computing, pp. 191–202. Springer, Heidelberg (2005)
10. Bazan, J.G., Skowron, A., Świniarski, R.W.: Rough sets and vague concept approximation: From sample approximation to adaptive learning. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 39–62. Springer, Heidelberg (2006)
11. Behnke, S.: Hierarchical Neural Networks for Image Interpretation. LNCS, vol. 2766. Springer, Heidelberg (2003)
12. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000), `http://portal.acm.org/citation.cfm?id=335191.335388`
13. Dietterich, T.G.: Hierarchical reinforcement learning with the MAXQ value function decomposition. Artificial Intelligence 13, 227–303 (2000)
14. Doherty, P., Łukaszewicz, W., Skowron, A., Szałas, A.: Knowledge Representation Techniques: A Rough Set Approach. Studies in Fuzziness and Soft Computing, vol. 202. Springer, Heidelberg (2006)
15. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, New York (2003)
16. Foster, I.: Service-oriented science. science 308, 814–817 (2005)
17. Gärdenfors, P.: The Geometry of Thought. MIT Press, Cambridge (2000)
18. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Elsevier, Morgan Kaufmann, San Francisco (2004)

19. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artif. Intell. Rev. 22(2), 85–126 (2004), `http://dx.doi.org/10.1023/B:AIRE.0000045502.10941.a9`
20. Jankowski, A., Peters, J., Skowron, A., Stepaniuk, J.: Optimization in discovery of compound granules. Fundamenta Informaticae 85(1-4), 249–265 (2008)
21. Jankowski, A., Skowron, A.: A wistech paradigm for intelligent systems. In: Peters, J.F., Skowron, A., Düntsch, I., Grzymała-Busse, J.W., Orłowska, E., Polkowski, L. (eds.) Transactions on Rough Sets VI. LNCS, vol. 4374, pp. 94–132. Springer, Heidelberg (2007)
22. Jankowski, A., Skowron, A.: Logic for artificial intelligence: The Rasiowa-Pawlak school perspective. In: Ehrenfeucht, A., Marek, V., Srebrny, M. (eds.) Andrzej Mostowski and Foundational Studies, pp. 106–143. IOS Press, Amsterdam (2008)
23. Jiang, F., Sui, Y., Cao, C.: Outlier detection using rough set theory. In: Slezak, D., Wang, M.D.I., Szczuka, G., Yao, Y. (eds.) 10th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Regina, Canada, pp. 79–87 (2005)
24. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of Artificial Intelligence Research 4, 227–303 (1996)
25. Kleinberg, J., Papadimitriou, C., Raghavan, P.: A microeconomic view of data mining. Data Mining and Knowledge Discovery 2, 311–324 (1998)
26. Knorr, E.M.: Outliers and data mining: Finding exceptions in data. Ph.D. thesis, University of British Columbia (2002)
27. Knorr, E.M., Ng, R.T.: Finding intensional knowledge of distance-based outliers. In: VLDB 1999: Proceedings of the 25th International Conference on Very Large Data Bases, pp. 211–222. Morgan Kaufmann, San Francisco (1999), `http://portal.acm.org/citation.cfm?id=671529`
28. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: Algorithms and applications. The VLDB Journal 8(3), 237–253 (2000)
29. McGovern, A.: Autonomous Discovery of Temporal Abstractions from Interaction with an Environment. University of Massachusetts, Amherst, MA, PhD thesis (2002)
30. Nguyen, H.S., Jankowski, A., Skowron, A., Stepaniuk, J., Szczuka, M.: Discovery of process models from data and domain knowledge: A rough-granular approach. In: Yao, J.T. (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation, pp. 1–30. IGI Global, Hershey, PA (submitted, 2008)
31. Nguyen, S.H., Bazan, J., Skowron, A., Nguyen, H.S.: Layered learning for concept synthesis. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 187–208. Springer, Heidelberg (2004)
32. Nguyen, T.T.: Eliciting domain knowledge in handwritten digit recognition. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 762–767. Springer, Heidelberg (2005)
33. Nguyen, T.T., Paddon, C.P.W.D.J., Nguyen, S.H., Nguyen, H.S.: Learning sunspot classification. Fundamenta Informaticae 72(1-3), 295–309 (2006)
34. Oliveira, L., Sabourin, R., Bortolozzi, F., Suen, C.: Feature selection using multiobjective genetic algorithms for handwritten digit recognition. In: Proc. 16th International Conference on Pattern Recognition (ICPR 2002), pp. 568–571. IEEE Computer Society Press, Los Alamitos (2002)
35. Pal, S.K., Polkowski, L., Skowron, A. (eds.): Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies. Springer, Berlin (2004)

36. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. System Theory, Knowledge Engineering and Problem Solving, vol. 9. Kluwer Academic Publishers, Dordrecht (1991)
37. Pawlak, Z., Skowron, A.: Rudiments of rough sets; Rough sets: Some extensions; Rough sets and boolean reasoning. Information Sciences 177(1), 3–27, 28–40, 41–73 (2007)
38. Pedrycz, W. (ed.): Granular Computing. Physica-Verlag, Heidelberg (2001)
39. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. John Wiley & Sons, New York (2008)
40. Peters, J.F.: Rough ethology: Towards a biologically-inspired study of collective behavior in intelligent systems with approximation spaces. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 153–174. Springer, Heidelberg (2004)
41. Peters, J.F.: Near sets: General theory about nearness of objects. Applied Mathematical Sciences 1(53), 2609–2629 (2007)
42. Peters, J.F.: Discovery of perceptually near information granules. In: J. Yao (ed.) Novel Developments in Granular Computing: Applications for Advanced Human Reasoning and Soft Computation. Information Science Reference, Hersey, NY (to appear, 2008)
43. Peters, J.F., Henry, C.: Reinforcement learning with approximation spaces. Fundamenta Informaticae 71, 323–349 (2006)
44. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
45. Polkowski, L., Skowron, A.: Rough mereology: A new paradigm for approximate reasoning. International Journal of Approximate Reasoning 51, 333–365 (1996)
46. Polkowski, L., Skowron, A.: Constructing rough mereological granules of classifying rules and classifying algorithms. In: Bouchon-Meunier, B., Gutierrez-Rios, J., Magdalena, L., Yager, R.R., Kacprzyk, J. (eds.) Technologies for Constructing Intelligent Systems I, pp. 57–70. Physica-Verlag (2002)
47. Rissanen, J.: Minimum-description-length principle. In: Kotz, S., Johnson, N. (eds.) Encyclopedia of Statistical Sciences, pp. 523–527. John Wiley & Sons, New York (1985)
48. Skowron, A.: Approximate reasoning in distributed environments. In: Zhong, Liu (eds.) [65], pp. 433–474
49. Skowron, A.: Rough sets in perception-based computing. In: Pal, S.K., Bandyopadhyay, S., Biswas, S. (eds.) PReMI 2005. LNCS, vol. 3776, pp. 21–29. Springer, Heidelberg (2005)
50. Skowron, A., Stepaniuk, J.: Tolerance approximation spaces. Fundamenta Informaticae 27, 245–253 (1996)
51. Skowron, A.: Information granules and rough-neural computing. In: Pal, S.K., Polkowski, L., Skowron, A. (eds.) Rough-Neural Computing: Techniques for Computing with Words. Cognitive Technologies, pp. 43–84. Springer, Heidelberg (2003)
52. Skowron, A., Stepaniuk, J.: Rough sets and granular computing: Toward rough-granular computing. In: Pedrycz, et al. (eds.) [39], pp. 425–448.
53. Skowron, A., Stepaniuk, J., Peters, J., Swiniarski, R.: Calculi of approximation spaces. Fundamenta Informaticae 72(1-3), 363–378 (2006)
54. Skowron, A., Stepaniuk, J., Peters, J.F.: Rough sets and infomorphisms: Towards approximation of relations in distributed environments. Fundamenta Informaticae 54(2–3), 263–277 (2003)

55. Skowron, A., Synak, P.: Complex patterns. Fundamenta Informaticae 60(1-4), 351–366 (2004)
56. Ślęzak, D.: Approximate entropy reducts. Fundamenta Informaticae 53(3–4), 365–387 (2002)
57. Stone, P.: Layered Learning in Multi-Agent Systems: A Winning Approach to Robotic Soccer. The MIT Press, Cambridge (2000)
58. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. The MIT Press, Cambridge (1998)
59. Van Wezel, W., Jorna, R., Meystel, A.: Automated Planning: Theory and Practice. John Wiley & Sons, Chichester (2006)
60. Vapnik, V.: Statisctical Learning Theory. John Wiley & Sons, Chichester (1998)
61. Vapnik, V.N.: The Nature of Statistical Learning Theory (Information Science and Statistics). Springer, Heidelberg (1999)
62. Zadeh, L.A.: A new direction in AI - toward a computational theory of perceptions. AI Magazine 22(1), 73–84 (2001)
63. Zadeh, L.A.: From imprecise to granular probabilities. Fuzzy Sets and Systems 154(3), 370–374 (2005)
64. Zadeh, L.A.: Generalized theory of uncertainty (GTU)-principal concepts and ideas. Computational Statistics and Data Analysis 51, 15–46 (2006)
65. Zhong, N., Liu, J. (eds.): Intelligent Technologies for Information Analysis. Springer, Heidelberg (2004)

# Integrative Levels of Granularity

Yiyu Yao

Department of Computer Science, University of Regina, Regina, Saskatchewan,
Canada S4S 0A2
yyao@cs.uregina.ca

**Abstract.** In their book, *Granular Computing: An Introduction*, Bargiela and Pedrycz present a view that granular computing is an emerging conceptual and computing paradigm of information processing. A central notion is an information-processing pyramid with multiple levels. Different levels involve different types of processing. The lowest level concerns numeric processing, the intermediate level concerns larger information granules, and the highest level concerns symbol-based processing. This chapter examines the notion of integrative levels of granularity as a basis of granular computing. The notion of levels had been studied extensively in different branches of sciences and different fields of computer sciences. By extracting a set of common features and principles of integrative levels of granularity, the triarchic theory of granular computing is developed.

**Keywords:** Integrative levels, granular computing triangle, triarchic theory of granular computing, structured thinking, structured problem solving, structured information processing, multilevel, multiview, granular structures.

## 1 Introduction

Granular computing is a multi-disciplinary and cross-disciplinary study (see, for examle, Bargiela and Pedrycz [3, 5], Inuiguchi et al. [20], Keet [21,22], Nguyen et al. [34], Lin et al. [27], Pawlak [38], Pedrycz et al. [39], Yao, JT [53], Yao [54], Zadeh [63], Zhang and Zhang [64]), concerning problem solving and information processing with multiple levels of granularity [55-59]. Granular computing may be viewed as human-inspired paradigms of computing and information processing, as well as their applications in the design and implementation of intelligent information systems [5, 61].

In their book, *Granular Computing: An Introduction*, Bargiela and Pedrycz [3] promote granular computing as an emerging conceptual and computing paradigm of information processing. The working principles of granular computing are explained based on an information-processing pyramid with multiple levels. Different levels involve different types of processing. The lowest level concerns numeric processing, the intermediate level concerns larger information granules, and the highest level concerns symbol-based processing. From this conceptual framework, we can identify two important notions, namely, granules and granular structures. Granules are elements and units that build up levels and granular structures are levels partially ordered based on their granularity. The formal representation of a granular structure as a multilevel hierarchical structure is based on further results from systems science, artificial intelligence and computer programming [57-59].

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 31–47.
springerlink.com

This chapter covers two aspects of granular computing. In Sections 2 to 4, we show that the notion of levels plays a fundamental role in many branches of sciences. A survey on many different interpretations and uses suggests that the concept of integrative levels of granularity may serve as a basis of granular computing. In Section 5, we briefly discuss the triarchic theory of granular computing that is centered around granular structures.

## 2 Integrative Levels

In this section, we argue that levels and associated multilevel hierarchical structures are common words of languages used in a wide spectrum of disciplines. A few important features of levels are examined.

### 2.1 Universality of Levels

The notions of levels and associated multilevel hierarchical structures are perhaps some of the most fundamental concepts and tools that we use to describe, represent, analyze and understand ourselves, reality and our relations to reality [1, 8, 11, 35, 41, 46, 57, 62]. As pointed out by Conger [8], the interpretations of differences of level range from the literal meanings to the various metaphorical meanings. The term "levels" seems to be a universal concept that has been widely used in philosophy and virtually all branches of natural and social sciences.

A (metaphysical) level, as defined by Conger [8], is "a class of structures or processes which are distinguishable from others as being either higher or lower." The terms "higher" and "lower" may denote various spatial, valuational, logical and developmental differences. Independent of any particular interpretation, such a higher-lower relation enables us to order levels, and hence to produce a multilevel hierarchical structure called a hierarchy.

In June 2008, we performed Google searches using a dozen phrases involving "levels." Table 1 summarizes the results about the number of hits of different phrases. Several observations can be made from a closer examination of various usages of levels and hierarchical structures.

Firstly, the notion of "levels" seems to be universally applicable to many different disciplines. The numbers of hits of various phrases containing "levels" range from several thousands to a few millions. Levels and hierarchical structures are used in virtually every branch of science and our daily life. Secondly, there seems to be a common understanding of levels, although slightly different interpretations exist in different fields. Levels may have either an objective or a subjective interpretation. The former reflects the intrinsic nature of reality; the latter reflects our cognitive understanding of reality. Levels are used to describe, organize and interpret things for the purposes of simplicity and clarity. Levels are sometimes used to denote a particular position on a scale, as reflected by levels of skill and levels of intelligence. Thirdly, levels are associated with the dual properties of separation and integration, and hence the term "integrative levels" and "integrated levels" are widely used. Levels generally imply a separation of things, with each level focusing on a particular aspect. Levels can be ordered partially to form a hierarchical or nested structure. That is,

**Table 1.** Google search results of various uses of levels

| Phrase | Hits |
|---|---|
| "levels of ability" | 1,300,000 |
| "levels of abstraction" | 1,030,00 |
| "levels of analysis" | 684,000 |
| "levels of business" | 163,000 |
| "levels of complexity" | 2,130,000 |
| "levels of comprehension" | 142,000 |
| "levels of cognition" | 35,100 |
| "levels of consciousness" | 216,000 |
| "levels of control" | 222,000 |
| "levels of description" | 86,100 |
| "levels of detail" | 333,000 |
| "levels of discovery" | 2,710,000 |
| "levels of evidence" | 113,000 |
| "levels of experience" | 309,000 |
| "levels of function" | 983,000 |
| "levels of government" | 2,800,000 |
| "levels of granularity" | 143,000 |
| "levels of intelligence" | 106,000 |
| "levels of interpretation" | 565,000 |
| "levels of intuition" | 14,500 |
| "levels of knowledge" | 207,000 |
| "levels of measurement" | 1,600,000 |
| "levels of observation" | 27,300 |
| "levels of organization" | 468,000 |
| "levels of perception" | 39,600 |
| "levels of processing" | 226,000 |
| "levels of reality" | 171,000 |
| "levels of reasoning" | 425,000 |
| "levels of representation" | 932,000 |
| "levels of skill" | 361,000 |
| "levels of strategy" | 1,220,000 |
| "levels of thinking" | 168,000 |
| "levels of thought" | 32,000 |
| "levels of understanding" | 245,000 |

many levels can be integrated to form a whole. A level can be further divided into sub-levels and many levels can be combined into one level, depending on our point of observation. One needs to study a level in the context of other levels. Fourthly, in theory there may be an arbitrary large number of many levels. However, most common uses of levels normally are within ten. Table 2 shows the numbers of hits on searching the numbers of levels commonly used. It can be seen that the most used numbers of levels are between two to four, with a peak at three. As the number of levels increases, the number of hits decreases. In other words, our understanding of reality is typically at a few levels, instead of a large number of levels.

The numbers of hits in Tables 1 and 2 should be read qualitatively. For example, the "2" in the phrase "2 levels of" may not actually mean that two levels are in fact used. Nevertheless, the observations from the tables are valid at a qualitative level. In

**Table 2.** Google search results of the numbers of levels

| Phrase | Hits | Phrase | Hits |
|---|---|---|---|
| “two levels of” | 3,510,000 | “2 levels of” | 636,000 |
| “three levels of” | 4,130,000 | “3 levels of” | 1,330,000 |
| “four levels of” | 1,460,000 | “4 levels of” | 477,000 |
| “five levels of” | 754,000 | “5 levels of” | 557,000 |
| “six levels of” | 284,000 | “6 levels of” | 191,000 |
| “seven levels of” | 195,000 | “7 levels of” | 141,000 |
| “eight levels of” | 122,000 | “8 levels of” | 146,000 |
| “nine levels of” | 40,300 | “9 levels of” | 79,900 |
| “ten levels of” | 60,500 | “10 levels of” | 99,400 |

particular, the trends obtained from the tables may be correct. For example, the phrase “levels of abstraction” is used more frequently than “the level of description.” People prefer a simple two- or three-level structure to other, more complex structures with many more levels. The results of Tables 1 and 2 perhaps deserve further attention and analysis, as they may enhance our understanding of granular computing where the notion of levels is of fundamental importance. In the next two subsections, we will examine in detail two aspects of the notion of levels.

A final note is that the term layers has also been widely used in place of levels. In fact, some authors use them interchangeably [41]. Like multilevel approaches, multilayer methodologies have been extensively studied. For example, a Google search of “multilevel” produces about 4.6 million hits; a search of “multilayer” produces about 4.4 million hits (searches done in November, 2008). In some sense, the term layers suggests a kind of total, linear ordering, namely, one layer on top of another. For levels, we only require a partial ordering. Studies on layers further demonstrate the universality of levels, as both of them roughly represent the same thing. The study of granular computing can also draw results from studies of layer-based approaches and multilayer methodologies. In some situations, the term layers may be intuitively more appealing.

## 2.2 Objective and Subjective Views of Levels

Levels and hierarchical structures are used to represent both reality and our perception and conceptualization of reality. Pattee [37] suggests that hierarchical systems may be characterized by the requirement of levels of description and the requirement of levels of structure. The requirement of levels of structure captures the inherent nature of a complex system, and the requirement of levels of description captures our understanding of the complex system. Poli [41] makes a similar distinction between the levels of reality and the levels of description. “The levels of reality have a strictly ontological valence, while those of description have a strictly epistemological one” [41]. A critical question put forward by Young [62], based on several other studies, is that of “whether hierarchies really exist or are simply a fiction of organizational convenience.” These studies suggest two extreme views for the interpretation of levels and a hierarchy, dealing with both the objective nature and the subjective nature of a hierarchy [22, 57]. Verdier [50] summarizes the two extremes as “the proponents of a hierarchy that is to be discovered” and “the proponents of an elaboration of a hierarchy by researchers.” In other words, the objective view is based on a position that the structural levels of matter are determined by the entirely objective laws of nature. Examples of this view include the levels of organization, the levels of control, and many more. The subjective view focuses on the human subjective multilevel understanding of reality. A hierarchy is formulated and built by the levels of description of our choice, which is based on our understanding through laws of the nature and the results of our observations. Examples of this view include the levels of description, the levels of representation, the levels of analysis, and many others.

At the same time, it may be not so easy to separate reality and our perception and understanding of reality, as described by the two views. It may be argued that we adopt the corresponding levels of description to reflect reality. In other words, our descriptions, in the form of hierarchical structures, merely mirror reality. For example, Hawkins [18] proposes that the human brain can be interpreted as a hierarchical structure that stores a model of the hierarchical structure of the real world. The real world's nested structure is mirrored by the nested structure of our cortex. Many of the phrases in Table 1 in fact reflect both the objective and subjective nature of levels.

The objective view on the existence of multilevel hierarchical structures may be explained in an evolutionary framework of complex systems proposed by Simon [47]. It is suggested that a hierarchy emerges almost inevitably through evolutionary processes for reasons of efficiency and stability. Systems can be quickly evolved to hierarchical structures and such structures are stable.

The subjective view on our imposition of hierarchical structures on reality may be explained based on the Miller's [29] finding about the limits of human information processing capacity. Our short-term memory holds around seven units of information. In order to cope with a large amount of information, the chunking principle is applied so that individual pieces of information are chunked together to form one larger unit. One may successively obtain a sequence of chunks so that the number of units in each level is within the capacity of the short-term memory. This process of chunking leads naturally to a hierarchical structure. Our hierarchical thinking is determined by our limited capacity to process information. It is not surprising that hierarchical structures are used universally. For example, hierarchical structures are used in our study of languages and knowledge, as well as in our reading and writing [15, 31, 60]). It have also been argued by many authors that human beings consistently search for order and human inclination to assert order may lead to the conception of hierarchy [36, 62].

Both the objective and the subjective natures of levels are well discussed in systems science, where hierarchy is a central concept [1, 46, 49]. On the one hand, it is assumed that "[t]he Universe is a hierarchy of systems; that is, simple systems are synthesized into more complex systems from subatomic particles to civilizations" [49]. The concept of hierarchy is a universal principle existing in natural, conceptual and man-made systems. Real world complex systems tend to organize hierarchically. On the other hand, it is also recognized that the word system "does not refer to existing things in the real world but rather to a way of organizing our thought about the real world" [49]. The study of systems is based the integration of the two views.

### 2.3 Separation and Integration of Levels

Studies on levels normally consider two related issues, namely, separation and integration. According to Novikoff [35], "The concept of integrative levels recognizes as equally essential for the purpose of scientific analysis both the isolation of parts of a whole and their integration into the structure of the whole." The two aspects are interwoven together in a hierarchy.

The separation of levels relies on two fundamental notions: loose coupling of parts in nearly-decomposable systems [47] and approximate knowledge [6]. On the one hand, it is important to realize that reality is a web in which everything is connected to everything else [6], and nature does not provide a picture where each level is clearly separated from the others. On the other hand, it is equally important to note that some things are more connected than others. We can explore the property of loose coupling in so-called nearly-decomposable systems to form various levels. Since such a separation of levels usually ignores subtle and small differences between individuals and their weak connections to others, the resulting multilevel hierarchical structures are approximations of reality. The knowledge obtained is in turn approximate. Nevertheless, such approximate knowledge is accurate and good enough for many practical purposes. The separation of levels thus gains in simplicity and clarity at the expense of accuracy.

Integration of levels is based on their interdependency and granularity. A level does not exist without its higher and/or lower levels. An ordering of levels is usually defined by the granularity of these levels. That is, different levels in general represent levels of differing complexity. Although a higher level depends on its lower levels, it has its unique properties that cannot be induced from lower levels. With integrative levels, we can easily shift our attention between different levels. By focusing on a particular level, we may study a specific aspect of reality.

Levels and hierarchies are the results of both separation and integration. Without separation, it is impossible to have levels; without integration, hierarchies do not exist. Levels are separated so that we can concentrate on a particular level at a specific point of time; levels are integrated so that we can observe the inter-working of all levels in a hierarchy. In a hierarchy, we can study the interaction of levels. A hierarchy allows both analytical thinking through separation and synthetical thinking through integration. Separation and integration may therefore be viewed as two sides of the same coin.

The separation and integration of levels offer two methods for constructing and interpreting a hierarchy: the top-down methods and the bottom-up methods. The bottom-up methods may be explained based an evolutionary framework of systems, from lower levels to higher levels. In the context of biology, Novikoff [35] suggests that new a level of complexity emerges from lower levels through organization and integration of units. The wholes on a lower level become parts on a higher level. Conger [8] discusses three issues in the development of later levels from earlier levels, namely, "(1) integration, or creative synthesis, (2) combining relations, or mutuality of relations, and (3) emergence of new qualities." The top-down approaches offer a good choice for representation, description, and understanding. Hierarchical thinking appears to be natural to many of us. When a system or phenomenon is explained from a skeleton to details, or from using general terms to using specific terms, it is much easier for us to understand. An understanding on one level makes an understanding on another level feasible.

A good example involving both bottom-up and top-down approaches is the writing process given by Flower and Hayes [12, 13]. In the phase of idea generation, one may build the thesis of an article through a bottom-up approach, where scattered points, facts, and ideas are progressively synthesized into a whole. In writing up the article, a top-down approach is used, where the thesis is broken into parts and these parts are further broken into sub-parts, based on a scheme represented by an ideas tree. An article is a product with multiple levels of detail, consisting of the title, the headings of the sections and subsections, paragraphs and individual sentences [60]. Similarly, the reading process of such constructed articles also involves a multiple level understanding [15]. The integrative understanding may be explained by hermeneutic circle, namely, "our understanding of the parts hinges on our understanding of a larger whole, which, again, can only be understood on the basis of the parts" [44].

## 3 A Short Survey on Studies of Levels

In this section, we briefly review studies of levels. It is not our intent to provide a complete survey, but a set of examples that are pertinent to our understanding and formulation of granular computing. In fact, from Table 1 we can easily see that a complete survey on all uses of levels is almost an impossible task.

### 3.1 Hierarchies in Systems Theory

In his book, *General Systems Theory, Ideas and Applications*, Skyttner [49] reviews more than a dozen systems theories and points out that all of them share a set of common properties. All of them (except one) are formulated based on hierarchies of both complexity and size. In addition, such hierarchical structures exist at all levels and on all scales. The theory of hierarchy is central to the general systems theory and some authors refer the former as a dialect of the latter [2].

Detailed descriptions of various levels in each of the systems theories are given in the Skyttner's book [49]. The properties or laws of levels have been studied and stated by many authors [2, 11, 35, 41, 42, 51, 52]. The following list gives a few of them:

1. Levels are populated by entities whose properties and interaction determines the level in question. Levels are formed based on laws of nature on the one hand and based on our cognitive understanding of reality on the other hand.
2. Levels represent both a separation of wholes into parts and an integration of parts into wholes. Each level is relatively autonomous and complete. It is possible to study each level within itself, as well as in the context of other levels. All levels are also integrated; a disturbance introduced at any one level reverberates at all other levels.
3. Levels are both continuous and discrete. New levels can always emerges from older levels in a continuous evolution and there may not exist a clear line that separates one level from another. At the same time, it is possible to identify various individual levels as the focal points of discussion.
4. Levels are ordered partially based on their corresponding complexity. At any given level, its mechanism lies at the level below and its purpose at the level above. A higher level normally has a smaller population of instances.
5. Each level depends on, and organizes, the level or levels below it. Each level has its emergent properties that cannot be deduced from lower levels. A level governs and controls its lower levels.

6. The knowledge of the lower level is necessary for a full understanding of the higher level; and yet it is impossible to predict the behavior of the high level based on such knowledge.

This is not a complete list and more properties may found in the given references. These properties are most relevant to our discussion on granular computing.

### 3.2 Levels in Cognitive Science and Psychology

The concept of levels has been considered in cognitive science and psychology in many different forms. We focus mainly on the human acquisition, processing and utilization of knowledge at multiple levels.

In characterizing human knowledge, one needs to consider two topics, namely, context and hierarchy [40, 48]. Knowledge is contextual and hierarchical. A context in which concepts are formed provides meaningful interpretation of the concepts. Knowledge is organized in a tower or a partial ordering. The base-level, or first-level, concepts are the most fundamental concepts, and higher-level concepts depend on lower-level concepts. Level thinking is of fundamental importance in the understanding, representation, organization and synthesis of data, information, and knowledge. Such a structured organization of knowledge seems to be one way to get around the limited capacity of human information processing, which was shown by Miller [29] and discussed earlier.

Levels of processing theory, proposed by Craik and Lockhart [10], presents a model of human memory and information processing in memory. The theory is formulated on the basis that human "perception involves the rapid analysis of stimuli at a number of levels or stages." While the earlier stages, i.e., shallow levels, process physical or sensory features, the later stages, i.e., deep levels, are more concerned with pattern recognition and meaning extraction. This model of a hierarchy of processing stages reflects multiple levels of the depth of processing, where a greater depth implies a greater degree of semantic or cognitive analysis. A more plausible alternative to such a sequential progression, from shallow to deep, is a combination of both stimulus-driven bottom-up processing and conceptually driven top-down processing [9]. The latter processing pattern seems to be consistent with the properties of levels discussed earlier.

Hierarchically structured knowledge has also been extensively explored in learning and student instruction. In teaching problem solving in physics, Reif and Heller [45] state, "effective problem solving in a realistic domain depends crucially on the content and structure of the knowledge about the particular domain." Knowledge about physics in fact specifies concepts and relations between them at various levels of abstraction. Furthermore, the knowledge is organized hierarchically, with explicit guidelines specifying when and how this knowledge is to be applied. Effective instruction needs to make effective use of such hierarchically structured knowledge. Posner [43] suggests that, according to the cognitive science approach, to learn a new field is to build appropriate cognitive structures and to learn to perform computations that will transform what is known into what is not yet known.

Levels of organization, levels of representation, levels of processing and levels of understanding are some of the important notions relevant to the study of human learning, information processing, and problem solving. The notion of levels helps us to explain and articulate many human activities and behaviors.

## 3.3 Levels in Computer Science

The notion of levels is widely used in computer science to describe and study various computer systems and concepts. With the introduction of levels, many concepts in computer science become easier to explain and understand. A few examples are discussed in this section.

In his work on vision, Marr [28] argues that a full understanding of an information processing system involves explanations at various levels. He proposes a three-level framework with each level addresses a particular of issues, which is quoted here:

1. Computational theory: What is the goal of the computation, why is it appropriate, and what is the logic of the strategy by which it can be carried out?
2. Representation and algorithm: How can this computational theory be implemented? In particular, what is the representation for the input and output, and what is the algorithm for the transformation?
3. Hardware implementation: How can the representation and algorithm be realized physically?

It can be said that each level addresses a different type of question, from abstract to concrete. The three levels are both dependent and independent.

Two basic notions, representation and process, are used to explain the three-level framework. The representation deals with the explicit forms of entities or types of information, the process deals with the operations on entities. The most abstract level deals with what the process does and why. One builds a theory that explains internal working principles of the process, and defines the operations by specifying constraints that must be satisfied by the process. The second level deals with the realization of the process in an abstract way. One needs to choose a representation for the input and for the expected output of the process, and to specify an algorithm for the transformation from input to output. The choices of representation and algorithm are closely tied together. There usually exist many alternative representations. For a given representation, there are also many possible algorithms. A representation and an algorithm should be chosen so that advantages of the representation are fully exploited by the algorithm and, at the same time, the disadvantages of the representation are avoided. The third level deals with the physical realization of the process. The devices that physically realize a process may not be unique. The advances in technologies imply that a same process may be implemented again with the invention of new physical devices.

Investigations at the computational theory level are independent of representations, and investigations at the representation and algorithm level is independent of physical devices. The levels are ordered and interpreted as levels of abstraction. The representation and algorithm level can also be named as logical implementation level, and the hardware implementation level as the physical implementation level.

In developing a theory of the nature of knowledge and representation, Newell [32, 33] introduces the concept of knowledge level. He views the nature of knowledge as the medium of a system level that lies above the symbol or program level. This is summarized by the so-called Knowledge Level Hypothesis [32]: “There exists a distinct computer systems level, lying immediately above the symbol level, which is characterized by knowledge as the medium and the principle of rationality as the law of behavior.” A framework of computer system levels thus consists of the device level, the circuit level, the logic level (with its two sublevels, combinatorial and sequential circuits, and the register-transfer level), the program or symbolic level, the knowledge level, and the configuration level (which is supported by the preceding three levels).

The systems levels as defined above are precise and operational. A level consists of a medium, components, laws of composition, and laws of behavior. Each level processes its medium based on the primitive processing provided by components, laws for assembling the components to form systems, and laws for determining the system behavior. There are additional characteristics of system levels. A given level may be implemented in many variant ways. Each level can be defined autonomously without reference to other levels, and each level can be reduced to, or realized by, the level below. Specification of a system at a particular level determines the system behavior at that level. The behavior of the total system is a combined result of local effects of all components. It is interesting to note some of these properties have also been extensively studied in systems science.

In the context of algorithm design, Foster [14] critically reviews and systematically compares various definitions and interpretations of the notion of levels. Three basic issues, namely, definition of levels, number of levels, and relationship between levels, are clarified. Three important points made by Foster are summarized as follows. First, levels are considered simply as descriptions or points of views and often for the purpose of explanation. Second, the number of levels is not fixed, but depends on the context and the purpose of description or explanation. Third, levels can be graphically represented as a vertical stack of planes. This multi-layered theory of levels captures two senses of abstraction. One is the abstraction in terms of concreteness and is represented by planes along the dimension from top to bottom. The other is the abstraction in terms of the amount of detail and can be modeled along another dimension from less detail to more detail on the same plane. The two senses of abstraction can be interpreted in terms of multiple hierarchies. The main hierarchy consists of levels, and each level is a hierarchy consisting of different levels of detail. For example, in the Marr’s three-level hierarchy, the logical implementation level may be a hierarchy consisting of logical implementations in various details. The abstraction in terms of detail is very useful in the implementation of information processing systems.

An excellent example of the effective use of levels is structured programming, characterized by top-down design and stepwise refinement. There is a huge body of literature on this topic. The following steps, taken from Ledgard et al. [25], are perhaps sufficient to illustrate the basic ideas:

1. Design in levels: A level consists of a set of modules. At higher levels, only a brief description of a module is provided. The details of the module are to be refined, divided into smaller modules, and developed in lower levels.
2. Initial language independence: The high-level representations at initial levels focus on expressions that are relevant to the problem solution, without explicit reference to machine and language dependent features.

3. Postponement of details to lower levels: The initial levels concern critical broad issues and the structure of the problem solution. The details such as the choice of specific algorithms and data structures are postponed to lower, implementation levels.
4. Formalization of each level: Before proceeding to a lower level, one needs to obtain a formal and precise description of the current level. This will ensure a full understanding of the structure of the current sketched solution.
5. Verification of each level: The sketched solution at each level must be verified, so that errors pertinent to the current level will be detected.
6. Successive refinements: Top-down programming is a successive refinement process. Starting from the top level, each level is redefined, formalized, and verified until one obtains a full program.

It is an easy task to apply the same principles elsewhere. For example, it has been suggested that the top-down approach is effective for developing, communicating and writing mathematical proofs [16, 23, 26]. The same principles can be applied in the preparation, organization, and writing of scientific articles [13, 60].

The operational feature of the notion of levels in computer science perhaps needs more emphasis. While the notion of levels is used in some disciplines as a tool for explanation, it is fully implemented in computer science in systems design, programming and many more. These actual implementations may provide necessary hints for implementing levels in related disciplines.

## 4 Implications of Integrative Levels to Granular Computing

The recent rise of granular computing may be compared with the rise of systems theory a few decades earlier, in terms of their philosophies, goals, scopes, and methodology [61]. The general systems theory attempts to discover and investigate structures and underlying principles common to most of natural and artificial systems [6, 17, 24, 49, 51]). The general systems theory is viewed as a science of sciences [49, 51] in an attempt to arrive at unity through diversity [17]. Similarly, research of granular computing attempts to discover and investigate structures and underlying principles common to most types of human problem solving [61]. As such, granular computing may be viewed as human-inspired computing and problem solving.

An important feature of human intelligence is that humans have many “Ways to Think” and can also create new “Ways to Think”, as suggested by Minsky [30]. Another feature is that humans form multiple representations of the world. This, in another way, motivates the study of granular computing. Granular computing can be viewed as a particular class of such “Ways to Think” that focuses on multiple levels of granularity. Furthermore, the notion of integrative levels may well serve the purpose of multiple representations.

Given this interpretation of granular computing, we can examine several important implications of the notion of integrative levels.

It is evident that the concept of integrative levels is very essential to human problem solving and has been used effectively time and again in many different disciplines. There seems to be a common set of interpretations, heuristics, principles and strategies of problem solving that are based on integrative levels. Unfortunately, these principles are scattered over many places in isolation without being synthesized into an integrated whole. They are normally explained with reference to discipline-specific knowledge and thus are buried deeply in minute details. Typically, the same principles are discussed in different languages and notations. In many occasions, we use these principles either implicitly or subconsciously because a formal documentation does not exist. This has led to the reinvention of the same principles time and again in the same or different fields.

The systems theory, to some extent, has resolved some of these problems. By introducing granular computing as a new field of study, we focus on a particular aspect. The notion of granularity is introduced to interpret the concept of integrative levels, and thus we have the notion of integrative levels of granularity. Each level is populated with granules of similar size or of similar nature. The levels of granularity may be interpreted as the levels of organization, levels of control, levels of complexity, levels of understanding, levels of description, levels of representation, levels of interpretation, levels of abstraction, levels of details, levels of processing and so on. The universality of levels implies that integrative levels of granularity may be used as a basis for granular computing.

The subjective view of levels suggests a hierarchy may only reflect our perception of reality and hence is only an approximation of reality. To remedy the shortcomings of such an approximation, it is necessary to consider many hierarchies in order to obtain multiple views of the same world [7, 30, 58]. With integrative levels of granularity, we consider granular structures that represent both multiple levels and multiple views. A single hierarchy gives one multilevel view of reality; many hierarchies give a mulitview description of reality [58]. Granular computing explores both of them.

The separation and integration of levels, together with the associated bottom-up and top-down methods, are related to the methodology of granular computing. The properties and laws of levels are useful in constructing various levels when applying granular computing principles in different domains. The implementation of levels in computer science offers more concrete ideas for applying ideas of granular computing.

## 5 The Triarchic Theory of Granular Computing

Once we accepted the notion of integrative levels of granularity as a basic concept of granular computing, we in fact emphasize a research direction that is dominated by structuredness. The study of granular computing depends crucially on granular structures that represent reality through multilevel and multiview.

The triarchic theory is a unified view that stresses the study of granular computing as a new field in its wholeness, rather than scattered pieces. Based on multiple level hierarchical structures, the triarachic theory integrates philosophical, methodological, and computational issues of granular computing as structured thinking, structured problem solving and structured information processing, respectively. A brief description of the theory is given in this section and more details can be found in [57-59, 61].

The core of the triarchic theory can be pictorially described by the granular computing triangle. The three vertices of the triangle represent the philosophical, methodological and computational perspectives.

**Philosophy: Granular computing as structured thinking.** The philosophy of granular computing offers a worldview characterized by different sized, interacting and hierarchically organized granules [19, 38, 63, 64]. This view of the world in terms of structures, as represented by multiple integrative levels, leads to a way of structured thinking, which is applicable to many branches of natural and social sciences. Broadly speaking, granular computing draws results from two complementary philosophical views about the complexity of real-world problems, namely, the traditional reductionist thinking and the more recent systems thinking. It combines analytical thinking for decomposing a whole into parts and synthetic thinking for integrating parts into a whole.

**Methodology: Granular computing as a general method of structured problem solving.** Granular computing promotes systematic approaches, effective principles, and practical heuristics and strategies that have been used effectively by humans for solving real-world problems. A central issue is the exploration of granular structures. This involves three basic tasks: constructing granular structures, working within a particular level of the structure, and switching between levels. We can formulate a set of principles to highlight the methodology of granular computing. For example, the principle of *multilevel granularity* emphasizes the effective use of a hierarchical structure. According to this principle, we must consider multiple representations at different levels of granularity. The principle of *multiview* stresses the consideration of diversity in modeling. We need to look at the same problem from many angles and perspectives. Once granular structures are obtained, we can apply other principles to work based on such structures. For example, the principle of *focused efforts* calls for attention on the focal point at a particular stage of problem solving; the principle of *granularity conversion* links the different stages in this process. The principle of *view switching* allows us to change views and to compare different views. These principles of granular computing have, in fact, been used extensively in different disciplines under different names and notations. Many principles of structured programming can be readily adopted for granular computing.

**Computation: Granular computing as a new paradigm of structured information processing.** Granular computing focuses on information processing methods based on the granular structures [4, 5]. The term computing needs to be understood in its broad meaning to include information processing in the abstract, in the brain and in machines. While information processing in the abstract deals with theories of computing without direct reference to their implementations, information processing in the brain and in machines represents the biological (natural) and the physical (artificial) implementations, respectively. Two related basic issues of computation are representations and processes (operations). Representation covers the formal and precise description of granules and granular structures. Processes may be broadly divided into the two classes: granulation and computation with granules. Granulation processes involve the construction of the building blocks and structures, namely, granules, levels, and hierarchies. Computation processes explore the granular structures. This involves two-way communication up and down in a hierarchy, as well as switching between levels.

The three perspectives of granular computing are connected and mutually support each other. A reviewer of this chapter points out the importance of studying the interactions of the three perspectives. The reviewer states, "In particular, a general picture illustrating interactions within the triangle would be helpful. … Speaking more about possible scenarios of interactions may turn out to be even more valuable than speaking about particular 'nodes' of the triangle." In some sense, the three perspectives can be interpreted as three levels of study, with the philosophical foundations supporting the methodological foundations, which in turn supports the computational implementations [57, 58]. It immediately follows that the arguments of separation and integration of levels can be directly applied to the separation and integration of three perspectives on granular computing.

With the separation of three perspectives, we emphasize the importance of philosophical and methodological studies that have played a relatively minor role so far. Granular computing offers a new way of thinking that may lead to a new set of problem-solving methods, or more precisely, a recasting of many existing methods in a new setting. Unfortunately, granular ways of thinking are not fully appreciated yet, due to some computational issues. Doubts on the potential of granular computing are commonly expressed at the computational level, as there still does exist a set of well-accepted algorithms or computational methods for granular computing. In [61], we argue that a lack of recent progresses in artificial intelligence may perhaps be explained by a viewpoint that paid little attention to human intelligence and how the brain works. A new school of thought is emerging that emphasizes the study of human brains and natural intelligence. If we view granular computing as human-inspired computing, we must study how humans solve problems by exploiting multiple levels of granularity. Consequently, we need to pay attention to the philosophical and methodological foundations of granular computing.

There is a top-down guiding role played by the three levels of granular computing. The philosophy of granular computing will guide us in searching for the right methodology; the methodology in turn can be applied in the design and implementation of granular-computing-based information systems. The separation of the three perspectives enables to us to ask the right questions and choose the right languages for granular computing at three levels. The philosophy of granular computing can be described in general terms. Its applications lead to two related classes of methodology, one for human problem solving and the other for machine problem solving. The former is more general than the latter; the latter is a specialization of the former. While methodology for humans may be qualitative and schematic, the methodology for machines must be precise and formal. At the next level, the methodology of granular computing is applied to concrete implementations. There is also a reverse bottom-up way of support. A study of granular-computing-based systems may offer new methodology, which in turn may help us in redefining our philosophical standing.

The three-level interpretation of granular computing is convenient, but of limited value. In general, the three perspectives are on the same footing and mutually support each other. That is, one node of the triangle supports, and is supported by, the other two; one cannot exist without the other two. This requires an integrated view that granular computing has three indispensable components. Any study that focuses only on some aspects may fail to realize the full potential of granular computing.

The triarchic theory puts granular computing research on a firm basis. In addition, the granular computing triangle recommends a research direction towards an interdisciplinary wholeness approach. That is, researchers in different disciplines may investigate different perspectives of granular computing and at the same time integrate their individual results.

## 6 Conclusion

The chapter examines a central notion of granular computing, namely, integrative levels of granularity. Two main features of integrative levels are discussed: the objective and subjective views of levels, and the separation and integration of levels. A survey on integrative levels and their basic properties, in several disciplines, suggests that integrative levels of granularity may serve as a basis for the study of granular computing. The triarchic theory of granular computing is briefly reviewed based on this notion.

Future progresses on the study of granular computing cannot be achieved based merely on investigations of concrete models or methods, namely, the computational perspective. Successful applications of granular computing may justify its existence; but they alone are far from enough. One needs to look at its foundations and roots [4]. In this regards, a conceptual framework, such as the triarchic theory, may be helpful.

## Acknowledgements

The author is grateful to an anonymous reviewer and Dr. Andrzej Bargiela for their constructive comments. In particular, the author would like to thank the reviewer for pointing out the needs for studying interactions between three nodes of the granular computing triangle.

## References

1. Ahl, V., Allen, T.F.H.: Hierarchy Theory, a Vision, Vocabulary and Epistemology. Columbia University Press (1996)
2. Allen, T.F.: A Summary of the Principles of Hierarchy Theory (accessed March 11, 2005), `http://www.isss.org/hierarchy.htm`
3. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Boston (2002)
4. Bargiela, A., Pedrycz, W.: The roots of granular computing. In: Proceedings of 2006 IEEE International Conference on Granular Computing, pp. 806–809 (2006)
5. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human-centered information processing. IEEE Transactions on Fuzzy Systems (to appear, 2008)
6. Capra, F.: The Web of Life. Anchor Books, New York (1997)
7. Chen, Y.H., Yao, Y.Y.: A multiview approach for intelligent data analysis based on data operators. Information Sciences 178, 1–20 (2008)
8. Conger, G.P.: The doctrine of levels. The Journal of Philosophy 22, 309–321 (1925)
9. Craik, F.I.M.: Levels of processing: past, present ...and future? Memory 10, 305–318 (2002)
10. Craik, F.I.M., Lockhart, R.S.: Levels of processing: a framework for memory research. Journal of Verbal Learning and Verbal Behavior 11, 671–684 (1972)

11. Feibleman, J.K.: Theory of integrative levels. The British Journal for the Philosophy of Science 5, 59–66 (1954)
12. Flower, L.: Problem-Solving Strategies for Writing. Harcourt Brace Jovabovich, Inc., New York (1981)
13. Flower, L.S., Hayes, J.R.: Problem-solving strategies and the writing process. College English 39, 449–461 (1977)
14. Foster, C.L.: Algorithms, Abstraction and Implementation: Levels of Detail in Cognitive Science. Academic Press, London (1992)
15. Frank, S.D.: Remember Everything You Read. Times Books, New York (1990)
16. Friske, M.: Teaching proofs: a lesson from software engineering. American Mathematical Monthly 92, 142–144 (1995)
17. Gray, W., Rizzo, N.D. (eds.): Unity Through Diversity, I & II. Gordon and Breach Science Publishers, New York (1973)
18. Hawkins, J., Blakeslee, S.: On Intelligence. Henry Holt and Company, New York (2004)
19. Hobbs, J.R.: Granularity. In: Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 432–435 (1985)
20. Inuiguchi, M., Hirano, S., Tsumoto, S. (eds.): Rough Set Theory and Granular Computing. Springer, Berlin (2003)
21. Keet, C.M.: A taxonomy of types of granularity. In: Proceeding of 2006 IEEE International Conference on Granular Computing, pp. 106–111 (2006)
22. Keet, C.M.: A Formal Theory of Granularity, PhD Thesis, KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy (2008) (accessed June 8, 2008), `http://www.meteck.org/files/AFormalTheoryOfGranularity_CMK08.pdf`
23. Lamport, L.: How to write a proof. American Mathematical Monthly 102, 600–608 (1995)
24. Laszlo, E.: The Systems View of the World: the Natural Philosophy of the New Developments in the Science. George Brasiller, New York (1972)
25. Ledgard, H.F., Gueras, J.F., Nagin, P.A.: PASCAL with Style: Programming Proverbs. Hayden Book Company, Inc., New Jersey (1979)
26. Leron, U.: Structuring mathematical proofs. American Mathematical Monthly 90, 174–185 (1983)
27. Lin, T.Y., Yao, Y.Y., Zadeh, L.A. (eds.): Data Mining, Rough Sets and Granular Computing. Physica-Verlag, Heidelberg (2002)
28. Marr, D.: Vision, A Computational Investigation into Human Representation and Processing of Visual Information. W.H. Freeman and Company, San Francisco (1982)
29. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review 63, 81–97 (1956)
30. Minsky, M.: The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human mind. Simon & Schuster Paperbacks, New York (2006)
31. Minto, B.: The Pyramid Principle: Logic in Writing and Thinking. Prentice Hall/Financial Times, London (2002)
32. Newell, A.: The knowledge level. Artificial Intelligence 18, 87–127 (1982)
33. Newell, A.: Reflections on the knowledge level. Artificial Intelligence 59, 31–38 (1993)
34. Nguyen, S.H., Skowron, A., Stepaniuk, J.: Granular computing: a rough set approach. Computational Intelligence 17, 514–544 (2001)
35. Novikoff, A.B.: The concept of integrative levels and biology. Science 101, 209–215 (1945)
36. Patrides, C.A.: Hierarchy and order. In: Dictionary of the History of Ideas, pp. 434–449. Scribners, New York (1973)
37. Pattee, H.H.: Unsolved problems and potential applications of hierarchy theory. In: Pattee, H.H. (ed.) Hierarchy Theory, The Challenge of Complex Systems, pp. 129–156. George Braziller, New York (1973)
38. Pawlak, Z.: Granularity of knowledge, indiscernibility and rough sets. In: Proceedings of 1998 IEEE International Conference on Fuzzy Systems, pp. 106–110 (1998)

39. Pedrycz, W., Skowron, A., Kreinovich, V. (eds.): Handbook of Granular Computing. Wiley, New York (2008)
40. Peikoff, L.: Objectivism: the Philosophy of Ayn Rand. Dutton, New York (1991)
41. Poli, R.: Levels. Axiomathes 9, 197–211 (1998)
42. Poli, R.: The basic problem of the theory of level of reality. Axiomathes 12, 261–283 (2002)
43. Posner, M.I. (ed.): Foundations of Cognitive Science. The MIT Press, Cambridge (1989)
44. Ramberg, B., Gjesdal, K.: Hermeneutics. The Stanford Encyclopedia of Philosophy (accessed June 14, 2005)
45. Reif, F., Heller, J.: Knowledge structure and problem solving in physics. Educational Psychologist 17, 102–127 (1982)
46. Salthe, S.N.: Evolving Hierarchical Systems, Their Structure and Representation. Columbia University Press (1985)
47. Simon, H.A.: The organization of complex systems. In: Pattee, H.H. (ed.) Hierarchy Theory, The Challenge of Complex Systems, pp. 1–27. George Braziller, New York (1973)
48. Simpson, S.G.: What is foundations of mathematics? (accessed November 21, 1996), `http://www.math.psu.edu/simpson/hierarchy.html`
49. Skyttner, L.: General Systems Theory, Ideas & Applications. World Scientific, Singapore (2001)
50. Verdier, B.: Hierarchy: a short history of a word in western thought. In: Pumain, D. (ed.) Hierarchy in Natural and Social Sciences, pp. 13–37. Springer, The Netherlands (2006)
51. von Bertalanffy, L.: An outline of general system theory. The British Journal for the Philosophy of Science 1, 134–165 (1950)
52. Wilson, T.D.: Philosophical foundations and research relevance: issues for information research. Journal of Information Science 29, 445–452 (2003)
53. Yao, J.T.: A ten-year review of granular computing. In: Proceedings of 2007 IEEE International Conference on Granular Computing, pp. 734–739 (2007)
54. Yao, Y.Y.: Granular computing: basic issues and possible solution. In: Proceedings of the 5th Joint Conference on Information Sciences, pp. 186–189 (2000)
55. Yao, Y.Y.: A partition model of granular computing. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 232–253. Springer, Heidelberg (2004)
56. Yao, Y.Y.: Granular computing. Computer Science (Ji Suan Ji Ke Xue) 31, 1–5 (2004b)
57. Yao, Y.Y.: Perspectives of granular computing. In: Proceedings of 2005 IEEE International Conference on Granular Computing, pp. 85–90 (2005)
58. Yao, Y.Y.: Three perspectives of granular computing. Journal of Nanchang Institute of Technology 25, 16–21 (2006)
59. Yao, Y.Y.: The art of granular computing. In: Kryszkiewicz, M., Peters, J.F., Rybinski, H., Skowron, A. (eds.) RSEISP 2007. LNCS, vol. 4585, pp. 101–112. Springer, Heidelberg (2007)
60. Yao, Y.Y.: Structured writing with granular computing strategies. In: Proceedings of 2007 IEEE International Conference on Granular Computing, pp. 72–77 (2007b)
61. Yao, Y.Y.: The rise of granular computing. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition) 20, 299–308 (2008)
62. Young, G.L.: Hierarchy and central place: some questions of more general theory. Geografiska Annaler, Series B, Human Geography 60, 71–78 (1978)
63. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–127 (1997)
64. Zhang, L., Zhang, B.: Theory and Application of Problem Solving – Theory and Application of Granular Computing in Quotient Spaces, 2nd edn. Tsinghua University Press, Beijing (2007)

# Affinities between Perceptual Granules: Foundations and Perspectives

James F. Peters[1] and Sheela Ramanna[2]

[1] Computational Intelligence Laboratory,
University of Manitoba, Winnipeg, Manitoba R3T 5V6 Canada
jfpeters@ee.umanitoba.ca

[2] Department of Applied Computer Science,
University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
s.ramanna@uwinnipeg.ca

**Abstract.** This chapter gives a concise overview of the foundations of a perceptual near set approach to the discovery of affinities between perceptual objects and perceptual granules that provide a basis for perceptual systems useful in science and engineering. A *perceptual object* is something perceptible to the senses or knowable by the mind. Perceptual objects that have the same appearance are considered to be perceptually near each other, i.e., perceived objects that have perceived affinities or, at least, similar descriptions. A perceptual granule is a set of perceptual objects originating from observations of the objects in the physical world. Near set theory provides a basis for observation, comparison and classification of perceptual granules. By considering nearness relations in the context of a perceptual system, it is possible to gauge affinities (nearness) perceptual objects. Two kinds of indiscernibility relations and a tolerance relation make it possible to define various nearness relations. Examples of near images as perceptual systems are presented. The main contribution of this chapter is the introduction of a formal basis for discovering affinities between perceptual information granules.

**Keywords:** Affinities, near sets, perceptual granule, tolerance relations.

## 1 Introduction

The basis for perceptual systems hearkens back to the original notion of a deterministic information system introduced by Zdzisław Pawlak [20]. A *perceptual system* is a real-valued, total, deterministic information system. A *perceptual object* is something perceptible to the senses or knowable by the mind. Examples of perceptual objects include observable organism behaviour, growth rates, soil erosion, events containing the outcomes of experiments such as energizing a network, testing digital camera functions, microscope images, MRI scans, and the results of searches for relevant web pages. Granulation can be viewed as a human way of achieving data compression and it plays a key role in implementing the divide-and-conquer strategy in human problem-solving. A comprehensive study of granular computing can be found in [1]. A perceptual granule is a set of perceptual objects originating from observations of the objects in the physical world. Formally, a *perceptual granule* is a finite, non-empty set containing sample

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 49–66.
springerlink.com

perceptual objects with common descriptions and a set probe functions representing perceptual object features.

Another means of discovering perceptual granules was suggested by Charles Darwin, who called attention to affinities that one can observe between different members of the same species. The proposed approach to discovering affinities between perceptual granules is analogous to what Charles Darwin did during the voyage of the H.M.S. Beagle during the 1830s, starting in 1831 and ending in 1836. That is, Darwin kept adding to his collection of specimens and eventually, in some cases, found affinities between a set of specimens of interest and his expanding set of specimens found during the voyage of the Beagle [3].

Near set theory provides a basis for observation, comparison and measuring affinities of perceptual granules. Near sets have a human-centric character. Sensed physical characteristics of perceptual objects are identified with object features. It is our mind that identifies relationships between object feature values to form perceptions of sensed objects [7]. Human perceptions can be quantified through the use of near sets by providing a framework for comparing objects based on object descriptions. Objects that have the same appearance (*i.e.*, objects with matching descriptions) are considered *perceptually near each other*. Sets are considered near each other when they have “things” (perceived objects) in common. Specifically, near sets facilitate measurement of similarities between perceptual objects based on feature values (obtained by probe functions) that describe the objects. This approach is similar to the way humans perceive objects (see, *e.g.*, [4]) and as such facilitates pattern classification systems.

Near sets originally grew out of a study of images [5, 28, 30, 33] either by considering single images containing near sub images or segmented images containing perceptually near pixel windows. Two kinds of indiscernibility relations and a tolerance relation make it possible to define various nearness relations. A weak tolerance relation is also defined in this chapter. This tolerance relation is very important in discovering near sets, since it defines tolerance classes relative to a threshold $\varepsilon$, rather than require strict equality of probe function values in the case of the indiscernibility relations. The underlying assumption made here is that human perception relies on a limited view of perceived objects to discover affinities between samples. For this reason, the discovery of near objects begins with the perception of one or more matching characteristics, not a complete set of matching characteristics. Finding a multitude of matches between perceptual objects is not considered in arriving at the discovery threshold in detecting affinities between objects, *i.e.*, in discovering near sets. This approach is in keeping with the original view of tolerance spaces as models for human vision [37].

The Pal entropy measure defined in [12] provides a useful basis for probe functions used in the search for perceptual granules that are, in some sense, near each other. Other forms of entropy introduced by Sankar Pal *et al.* can be found in [9, 13, 14, 15, 16, 17, 18]. It has been shown that perceptual near sets are a generalization of rough sets introduced by Zdzisław Pawlak during the early 1980s. That is, every rough set is a near set but not every near set is a rough set. In addition, it can be shown that fuzzy sets with non-empty cores are near sets. The connections between these three forms of sets are briefly discussed in this chapter. By way of an illustration, affinities between microscope images (as elements in perceptual systems) of various leaves of trees are briefly explored.

This chapter is organized as follows. Section 2 presents the basis for perceptual systems. Indiscernibility relations and a tolerance relation are introduced in Section 3. Three basic nearness relations are presented and illustrated in Section 3 accompanied by an illustration of near images in Section 4.2. Examples of rough near sets and fuzzy near sets are presented in Sections 5 and 6, respectively.

## 2 Perceptual Systems: An Overview

This section briefly presents the basis for perceptual systems that hearkens back to the original notion of a deterministic information system introduced by Zdzisław Pawlak [20] and elaborated in [10, 11].

### 2.1 Perceptual Object Descriptions

Perceptual objects are known by their descriptions. An *object description* is defined by means of a tuple of function values $\phi(x)$ associated with an object $x \in X$ (see Table 1). The important thing to notice is the choice of functions $\phi_i \in \mathscr{B}$ used to describe an object of interest. Assume that $\mathscr{B} \subseteq \mathbb{F}$ (see Table 1) is a given set of functions representing features of sample objects $X \subseteq O$ and $\mathbb{F}$ is finite. Let $\phi_i \in \mathscr{B}$, where $\phi_i : O \longrightarrow \mathbb{R}$. In combination, the functions representing object features provide a basis for an *object description* $\phi : O \longrightarrow \mathbb{R}^l$, a vector containing measurements (returned values) associated with each functional value $\phi_i(x)$ for $x \in X$, where $|\phi| = l$, i.e. the description length is $l$.

$$\textbf{Object Description:} \quad \phi(x) = (\phi_1(x), \phi_2(x), \ldots, \phi_i(x), \ldots, \phi_l(x)).$$

The intuition underlying a description $\phi(x)$ is a recording of measurements from sensors, where each sensor is modeled by a function $\phi_i$. Notice that all sensor values belong to the set of reals. That is, the perception of an object (*i.e.*, in effect, our knowledge about an object) depends on information gathered by our senses. The proposed approach to perception is feature-based and is similar to the one discussed in the introduction in [2].

**Table 1.** Description Symbols

| Symbol | Interpretation |
|---|---|
| $\mathbb{R}$ | Set of real numbers, |
| $O$ | Set of perceptual objects, |
| $X$ | $X \subseteq O$, set of sample objects, |
| $x$ | $x \in O$, sample object, |
| $\mathbb{F}$ | A set of functions representing object features, |
| $\mathscr{B}$ | $\mathscr{B} \subseteq \mathbb{F}$, |
| $\phi$ | $\phi : O \rightarrow \mathbb{R}^l$, object description, |
| $l$ | $l$ is a description length, |
| $i$ | $i \leq l$, |
| $\phi_i$ | $\phi_i \in \mathscr{B}$, where $\phi_i : O \longrightarrow \mathbb{R}$, probe function, |
| $\phi(x)$ | $\phi(x) = (\phi_1(x), \ldots, \phi_i(x), \ldots, \phi_L(x))$, description, |
| $\langle X, \mathbb{F} \rangle$ | $\phi(x_1), \ldots, \phi(x_{\lvert X \rvert})$, *i.e.*, perceptual information system. |

In this view, our senses are likened to probe functions, *i.e.*, mappings of sensations to values assimilated by the mind.

Let $X, Y \subseteq O$ denote sets of perceptual objects. Sets $X, Y \subseteq O$ are considered near each other if the sets contain perceptual objects with at least partial matching descriptions. A *perceptual object* $x \in O$ is something presented to the senses or knowable by the mind [8]. In keeping with the approach to pattern recognition suggested by Pavel [19], the features of an object such as contour, colour, shape, texture, bilateral symmetry are represented by probe functions. A *probe function* can be thought of as a model for a sensor. A probe makes it possible to determine if two objects are associated with the same pattern without necessarily specifying which pattern (classification). A detailed explanation about probe functions vs. attributes in the classification of objects is given in [26].

### 2.2 Perceptual Systems: Specialized Deterministic Systems

For representing results of a perception, the notion of a perceptual system is briefly introduced in this section. In general, an *information system* is a triple $S = \langle Ob, At, \{Val_f\}_{f \in At}\rangle$ where *Ob* is a set of objects, *At* is a set of functions representing either object features or object attributes, and each $Val_f$ is a value domain of a function $f \in At$, where $f : Ob \longrightarrow \mathscr{P}(Val_f)$, ($\mathscr{P}(Val_f)$ is a power set of $Val_f$) (see, *e.g.*, citePawlak1983). If $f(x) \neq \emptyset$ for all $x \in Ob$ and $f \in At$, then $S$ is *total*. If $card(f(x)) = 1$ for every $x \in Ob$ and $f \in At$, then $S$ is *deterministic*. Otherwise $S$ is *non-deterministic*. In the case, when $f(x) = \{v\}$, $\{v\}$ is identified with $v$. An information system $S$ is *real valued* iff $Val_f = \mathbb{R}$ for every $f \in At$. Very often a more concise notation is used: $\langle Ob, At\rangle$, especially when value domains are understood, as in the case of real valued information systems. Since we focus on sensed objects we consider each $f \in At$ to be a *probe functions*. Two examples of perceptual systems are given in Table 2 (see 3.1 for a discussion of the examples).

**Table 2.** Sample perceptual information systems

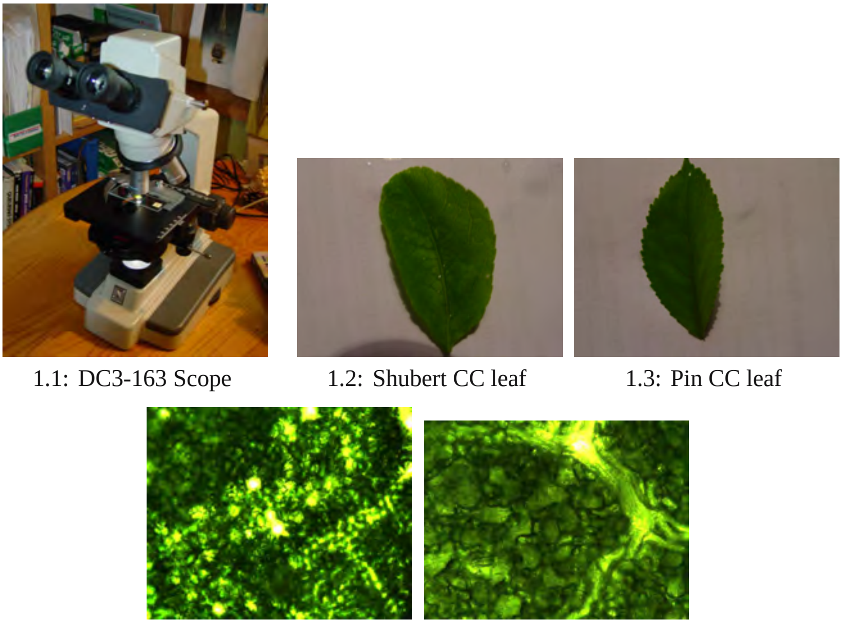
| Sys. 1 | | | | | | Sys. 2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **X** | $\phi_1$ | $\phi_2$ | $\phi_3$ | $\phi_4$ | | **Y** | $\phi_1$ | $\phi_2$ | $\phi_3$ | $\phi_4$ |
| $x_1$ | 0 | 1 | 0.1 | 0.75 | | $y_1$ | 0 | 2 | 0.2 | 0.01 |
| $x_2$ | 0 | 1 | 0.1 | 0.75 | | $y_2$ | 1 | 1 | 0.25 | 0.01 |
| $x_3$ | 1 | 2 | 0.05 | 0.1 | | $y_3$ | 1 | 1 | 0.25 | 0.01 |
| $x_4$ | 1 | 3 | 0.054 | 0.1 | | $y_4$ | 1 | 3 | 0.5 | 0.55 |
| $x_5$ | 0 | 1 | 0.03 | 0.75 | | $y_5$ | 1 | 4 | 0.6 | 0.75 |
| $x_6$ | 0 | 2 | 0.02 | 0.75 | | $y_6$ | 1 | 4 | 0.6 | 0.75 |
| $x_7$ | 1 | 2 | 0.01 | 0.9 | | $y_7$ | 0 | 2 | 0.4 | 0.2 |
| $x_8$ | 1 | 3 | 0.01 | 0.1 | | $y_8$ | 0 | 3 | 0.5 | 0.6 |
| $x_9$ | 0 | 1 | 0.5 | 0.1 | | $y_9$ | 0 | 3 | 0.5 | 0.6 |
| $x_{10}$ | 1 | 1 | 0.5 | 0.25 | | $y_{10}$ | 1 | 2 | 0.7 | 0.4 |
| | | | | | | $y_{11}$ | 1 | 4 | 0.6 | 0.8 |
| | | | | | | $y_{12}$ | 1 | 4 | 0.7 | 0.9 |
| | | | | | | $y_{13}$ | 1 | 1 | 0.25 | 0.01 |
| | | | | | | $y_{14}$ | 1 | 4 | 0.6 | 0.75 |

**Definition 1. Perceptual System**
*A* perceptual system $\langle O, \mathbb{F} \rangle$ *is a real valued total deterministic information system where $O$ is a non-empty set of* perceptual objects*, while $\mathbb{F}$ a countable set of* probe functions.

The notion of a perceptual system admits a wide variety of different interpretations that result from the selection of sample perceptual objects contained in a particular sample space $O$. Perceptual objects are known by their descriptions. For simplicity, we consider only small sets of probe functions in this chapter. The question of countable (denumerable) sets of probe functions is not within scope of this paper.

### 2.3 Sample Perceptual System

By way of an illustration, let $\langle P, \phi \rangle$ denote a perceptual system where $P$ is a set of microscope images and $\phi$ is a probe function representing luminance contrast[1], respectively. A sample Shubert choke cherry leaf and Native Pin choke cherry leaf are shown in Figures 1.2 and 1.3. The National Optical DC3-163 microscope in Fig. 1.1 was used to produce the magnified leaf-section images shown in Figures 1.4 and 1.5 with a lens that magnifies the size of an object by a factor of 40. Intuitively, if we compare colour, luminance contrast or sub image shapes, the microscope leaf images are similar. By



1.1: DC3-163 Scope

1.2: Shubert CC leaf

1.3: Pin CC leaf

1.4: Shubert CC slide

1.5: Pin CC slide

**Fig. 1.** Sample Percepts

[1] In digital images, luminance contrast can be controlled by converting irradiance (amount of light per unit area) into a grey value $g$ using a function $g(E) = E^{\gamma}$, where $E$ denotes irradiance level and luminance varies non-linearly with $\gamma$ typically having a value of 0.4 [6].

considering nearness relations in the context of a perceptual system, it is possible to classify sets of perceptual objects. A formal basis for the discovery of near sets is the focus of the remaining sections of this chapter.

# 3 Relations, Partitions and Classes

The basic idea in the near set approach to object recognition is to compare object descriptions. Sample perceptual objects $x,y \in O, x \neq y$ are near each other if, and only if $x$ and $y$ have similar descriptions. Similarly, sets $X,Y$ are perceptually near each other in the case where there is at least one pair of objects $x \in X, y \in Y$ that have similar descriptions. In this section, two kinds of indiscernibility relations and a tolerance relation are briefly introduced. These relations make it possible to define various nearness relations and make it possible to provide a formal foundation for near sets.

## 3.1 Indiscernibility and Tolerance Relations

Recall that each $\phi$ defines the description of an object (see Table 1). To establish a nearness relation, we first consider the traditional indiscernibility relation. Let $\mathscr{B} \subseteq \mathbb{F}$ denote a set of functions representing perceptual object features. The indiscernibility relation $\sim_{\mathscr{B}}$ introduced by Zdzisław Pawlak [20] is distinguished from weak indiscernibility $\underline{\bowtie}$ introduced introduced by Ewa Orłowska [10]. In keeping with the original indiscernibility relation symbol $\sim_{\mathbb{F}}$ [20], the symbol $\underline{\bowtie}$ is used to denote weak indiscernibility instead of the notation *wind* [10].

**Definition 2. Indiscernibility Relation**
*Let $\langle O, \mathbb{F} \rangle$ be a perceptual system. For every $\mathscr{B} \subseteq \mathbb{F}$ the indiscernibility relation $\sim_{\mathscr{B}}$ is defined as follows:*

$$\sim_{\mathscr{B}} = \{(x,y) \in O \times O \mid \ \forall \phi_i \in \mathscr{B} \,\text{.}\ \phi_i(x) = \phi_i(y)\}\,.$$

*If $\mathscr{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\sim_{\{\phi\}}$ we write $\sim_{\phi}$.*

**Example 1. Sample Partitions**
Let $\langle O_1, \mathbb{F}_1 \rangle$ denote perceptual system Sys. 1 with $O_1 = \{x_1, \ldots, x_9\}$, $\mathbb{F}_1 = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of probe functions from $\mathbb{F}_1$ are given in the lefthand side of table 2. Similarly, let $\langle O_2, \mathbb{F}_2 \rangle$ denote perceptual system Sys. 2 with $O_2 = \{y_1, \ldots, x_{14}\}$, $\mathbb{F}_2 = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of the probe functions from $\mathbb{F}_1$ are given in the righthand side of table 2. The perceptual systems $\langle O_1, \mathbb{F}_1 \rangle$, $\langle O_2, \mathbb{F}_2 \rangle$ have partitions (1) and (2-1.3) of the space of percepts defined by relations $\sim_{\mathbb{F}_1}$ and $\sim_{\mathbb{F}_2}$.

$$O_{1/\sim_{\mathbb{F}_1}} = \{\{x_1,x_2\},\{x_3\},\{x_4\},\{x_5\},\{x_6\},\{x_7\},\{a_8\},\{x_9\},\{x_{10}\}\}, \tag{1}$$

$$O_{2/\sim_{\mathbb{F}_2}} = \{\{y_1\},\{y_2,y_3,y_{13}\},\{y_4\},\{y_5,y_6\},\{y_7\},\{y_8,y_9\},\{y_{10}\}, \tag{2}$$

$$\{y_{11}\},\{y_{12}\},\{y_{14}\}\}. \tag{3}$$

If we consider only probe function $\phi_3$ relative to $O_1$, then we obtain, *e.g.*, several equivalence classes such as (4), each containing a pair of objects.

$$x_{1/\sim_{\phi_3}} = \{x_1, x_2\}, \tag{4}$$

$$x_{7/\sim_{\phi_3}} = \{x_7, x_8\}, \tag{5}$$

$$x_{9/\sim_{\phi_3}} = \{x_9, x_{10}\}. \tag{6}$$

Again, for example, if we probe $O_2$ with $\phi_3$, we obtain, *e.g.*, a number of multi-object classes such as the one in (7).

$$y_{2/\sim_{\phi_3}} = \{y_2, y_3, y_{13}\}, \tag{7}$$

$$y_{4/\sim_{\phi_3}} = \{y_4, y_8, y_9\}, \tag{8}$$

$$y_{5/\sim_{\phi_3}} = \{y_5, y_6, y_{11}, y_{14}\}, \tag{9}$$

$$y_{10/\sim_{\phi_3}} = \{y_{10}, y_{12}\}. \tag{10}$$

**Definition 3. Weak Indiscernibility Relation**
*Let $\langle O, \mathbb{F}\rangle$ be a perceptual system. For every $\mathscr{B} \subseteq \mathbb{F}$ the weak indiscernibility relation $\simeq_{\mathscr{B}}$ is defined as follows:*

$$\simeq_{\mathscr{B}} = \{(x,y) \in O \times O \mid\ \exists \phi_i \in \mathscr{B} \centerdot\ \phi_i(x) = \phi_i(y)\}\,.$$

*If $\mathscr{B} = \{\phi\}$ for some $\phi \in \mathbb{F}$, instead of $\simeq_{\{\phi\}}$ we write $\simeq_\phi$.*

**Example 2. Weak Indiscernibility Partitions**
Let $\langle O_1, \mathbb{F}_1\rangle$ denote perceptual system Sys. 1 with $O_1 = \{x_1, \ldots, x_9\}$, $\mathbb{F}_1 = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of probe functions from $\mathbb{F}_1$ are given in the lefthand side of table 2. Similarly, let $\langle O_2, \mathbb{F}\rangle$ denote perceptual system Sys. 2 with $O_2 = \{y_1, \ldots, y_{14}\}$, $\mathbb{F} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of the probe functions from $\mathbb{F}$ are given in the right-hand side of table 2. Let $X \subset O_1, X = \{x_1, x_9, x_{10}\}$ and $Y \subset O_2, Y = \{y_1, y_8, y_{10}, y_{11}, y_{12}\}$. Consider partitions $X_{/\simeq_{\phi_3}}$ and $Y_{/\simeq_{\phi_3}}$ given in (11) and (12), respectively.

$$X_{/\simeq_{\phi_3}} = \{\{x_1\}, \{x_9, x_{10}\}\}, \tag{11}$$

$$Y_{/\simeq_{\phi_3}} = \{\{y_1\}, \{y_8\}, \{y_{10}\}, \{y_{11}\}, \{y_{12}\}\}, \tag{12}$$

**Remark 1.** Notice that the class $\{x_1\} \in X_{/\cong_{\phi_3}}$ contains only a single object, since there is no other object in $x \in X$ such that $\phi_3(x_1) = \phi_3(x)$. Similarly, each of the classes in $Y_{/\cong_{\phi_3}}$ contains only a single object.

**Definition 4. Weak Tolerance Relation**
Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $\varepsilon \in \Re$ (reals). For every $\mathscr{B} \subseteq \mathbb{F}$ the weak tolerance relation $\cong_{\mathscr{B},\varepsilon}$ is defined as follows:

$$\cong_{\mathscr{B},\varepsilon} = \{(x,y) \in O \times O \mid\ \exists \phi_i \in \mathscr{B} \centerdot\ |\phi_i(x) - \phi_i(y)| \leq \varepsilon\}\,.$$

That is, in general, the relation $\cong_{\mathscr{B},\varepsilon}$ is reflexive and symmetric but not transitive. This relation is very important in discovering near sets, since it defines tolerance classes relative to a threshold $\varepsilon$, rather than require strict equality of probe function values in the case of the indiscernibility relations $\sim_{\mathscr{B}}$ and $\simeq_{\mathscr{B}}$ (see, *e.g.*, [30]).

**Remark 2. Special Case**
Notice that Def. 4 represents a special case. That is, in general, the sets $X$ and $Y$ represent sample sets of observations from distinct perceptual systems. In effect, it is possible to state a Proposition to this effect.

**Definition 5. Weak Tolerance Relation Between Sets of Perceptual Objects**
Let $P1 = \langle O_1, \mathbb{F}\rangle$ denote perceptual system P1. Similarly, let $P2 = \langle O_2, \mathbb{F}\rangle$ denote a second, distinct perceptual system. Also, let $\varepsilon \in \Re$. $P_1$ has a weak tolerance relation to $P_2$ if, and only if $O_1 \cong_{\mathbb{F},\varepsilon} O_2$.

**Definition 6. Weak Tolerance Relation on Perceptual Systems**
Let $Sys1 = \langle O_1, \mathbb{F}\rangle$ denote perceptual system Sys1. Similarly, let $Sys2 = \langle O_2, \mathbb{F}\rangle$ denote a second, distinct perceptual system with the same set of features $\mathbb{F}$. Let $\mathscr{B} \subseteq \mathbb{F}$ and choose $\varepsilon$. Then

$$Sys1 \cong_{\mathscr{B},\varepsilon} Sys1 \iff O_1 \cong_{\mathscr{B},\varepsilon} O_2.$$

**Example 3. Weak Tolerance**
Let $\langle O_1, \mathbb{F}\rangle$ denote perceptual system Sys. 1 with $O_1 = \{x_1, \ldots, x_9\}$, $\mathbb{F} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of probe functions from $\mathbb{F}$ are given in the lefthand side of table 2. Similarly, let $\langle O_2, \mathbb{F}\rangle$ denote perceptual system Sys. 2 with $O_2 = \{y_1, \ldots, y_{14}\}$, $\mathbb{F} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of the probe functions from $\mathbb{F}$ are given in the righthand side of table 2. Let $\varepsilon = 0.1$ for both perceptual systems. For example, let $\phi_3 \in \mathbb{F}_1$. The perceptual system $\langle O_1, \{\phi_3\}\rangle$ has tolerance classes (13), (14), (15) defined by relation $\cong_{\phi_{3,0.1}}$.

$$x_{1/\cong_{\phi_3,0.1}} = \{x_1, x_2, x_5, x_6, x_7, x_8\}, \tag{13}$$

$$x_{3/\cong_{\phi_3,0.1}} = \{x_3, x_4\}, \tag{14}$$

$$x_{9/\cong_{\phi_3,0.1}} = \{x_9, x_{10}\}. \tag{15}$$

For example, in $x_{3/\cong_{\phi_3,0.1}}$, we have

$$|\phi_3(x_3) - \phi_3(x_4)| = |0.05 - 0.054| \leq 0.1$$

Similarly, the perceptual system $\langle O_2, \{\phi_3\}\rangle$ has tolerance classes defined by relation $\cong_{\phi_{3,0.1}}$: (16), (17), (18), (19) .

$$y_{1/\cong_{\phi_3,0.1}} = \{y_1, y_2, y_3, y_{13}\}, \tag{16}$$

$$y_{4/\cong_{\phi_3,0.1}} = \{y_4, y_5, y_6, y_8, y_9, y_{11}, y_{14}\}, \tag{17}$$

$$y_{7/\cong_{\phi_3,0.1}} = \{y_7, y_4, y_8, y_9\}, \tag{18}$$

$$y_{10/\cong_{\phi_3,0.1}} = \{y_5, y_6, y_{10}, y_{11}, y_{12}, y_{14}\}, \tag{19}$$

For example, in $y_{7/\cong_{\phi_3,0.1}}$, we have

$$
\begin{aligned}
|\phi_3(y_7) - \phi_3(y_4)| &= |0.4 - 0.5| \leq 0.1,\\
|\phi_3(y_7) - \phi_3(y_8)| &= |0.4 - 0.5| \leq 0.1,\\
|\phi_3(y_7) - \phi_3(y_9)| &= |0.4 - 0.5| \leq 0.1,\\
|\phi_3(y_8) - \phi_3(y_9)| &= |0.5 - 0.5| \leq 0.1
\end{aligned}
$$

## 4 Nearness Relations

Three basic nearness relations are briefly presented and illustrated in this section.

**Definition 7. Nearness Relation** [34]
Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $X, Y \subseteq O$. The set $X$ is perceptually near to the set $Y$ ($X \bowtie_{\mathbb{F}} Y$), if and only if there are $x \in X$ and $y \in Y$ such that $x \sim_{\mathbb{F}} y$ (see Table 3).

**Table 3.** Relation Symbols

| Symbol | Interpretation |
|---|---|
| $\mathscr{B}$ | see Table 1, |
| $\varepsilon$ | $\varepsilon \in [0,1]$, |
| $\sim_{\mathscr{B}}$ | $\{(x,y) \mid f(x) = f(y)\ \forall f \in \mathscr{B}\}$, indiscernibility relation [20], |
| $\simeq_{\mathscr{B}}$ | weak indiscernibility relation [10], |
| $\cong_{\mathscr{B},\varepsilon}$ | weak tolerance relation, |
| $x_{/\sim_{\mathscr{B}}}$ | $x_{/\sim_{\mathscr{B}}} = \{y \in X \mid y \sim_{\mathscr{B}} x\}$, elementary set (class), |
| $O_{/\sim_{\mathscr{B}}}$ | $O_{/\sim_{\mathscr{B}}} = \{x_{/\sim_{\mathscr{B}}} \mid x \in O\}$, quotient set, |
| $\bowtie$ | nearness relation symbol, |
| $\underline{\bowtie}$ | weak nearness relation symbol, |
| $\underline{\underline{\bowtie}}$ | weak tolerance nearness relation symbol. |

**Example 4.** Consider the perceptual systems $\langle O_1, \mathbb{F}\rangle$, $\langle O_2, \mathbb{F}\rangle$ given in Table 2. From Example 2, we obtain

$$
\begin{aligned}
\mathscr{B} &= \{\phi_3\},\ \text{where } \phi_3 \in \mathbb{F},\\
X_{new} &= x_{9/\sim_{\phi_3}},\ \text{from Example 2},\\
&= \{x_9, x_{10}\},\\
Y_{new} &= y_{8/\sim_{\phi_3}}\\
&= \{y_4, y_8, y_9\},\\
X_{new}\ &\bowtie_{\phi_3}\ Y_{new},\ \text{since}\\
\phi_3(x_9) &= \phi_3(y_8) = 0.5
\end{aligned}
$$

**Definition 8. Weak Nearness Relation** [34]
Let $\langle O, \mathbb{F}\rangle$ be a perceptual system and let $X, Y \subseteq O$. The set $X$ is weakly near to the set $Y$ *within* the perceptual system $\langle O, \mathbb{F}\rangle$ ($X \underline{\bowtie}_{\mathbb{F}} Y$) iff there are $x \in X$ and $y \in Y$ and there is $\mathscr{B} \subseteq \mathbb{F}$ such that $x \simeq_{\mathscr{B}} y$. If a perceptual system is understood, then we say shortly that a set $X$ *is weakly near to* set $Y$ (see Table 3).

**Example 5.** Consider the perceptual systems $\langle O_1,\mathbb{F}\rangle$, $\langle O_2,\mathbb{F}\rangle$ given in Table 2.

$$\begin{aligned}
&\mathscr{B} = \{\phi_3\}, \text{ where } \phi_3 \in \mathbb{F},\\
&X = \{x_1,x_2,x_7,x_8,x_9,x_{10}\},\\
&Y = \{y_4,y_5,y_6,y_8,y_9,y_{11}\},\\
&X \underline{\bowtie}_{\phi_3} Y, \text{ since we can find } x \in X, y \in Y \text{ where } x \simeq_{\phi_3} y, \textit{ e.g.},\\
\phi_3(x_9) = \phi_3(y_8) = 0.5.
\end{aligned}$$

**Definition 9. Weak Tolerance Nearness Relation** [30]
Let $\langle O,\mathbb{F}\rangle$ be a perceptual system and let $X,Y \subseteq O, \varepsilon \in [0,1]$. The set $X$ is perceptually near to the set $Y$ *within* the perceptual system $\langle O,\mathbb{F}\rangle$ ($X \underline{\underline{\bowtie}}_{\mathbb{F}} Y$) iff there exists $x \in X$, $y \in Y$ and there is a $\phi \in \mathbb{F}, \varepsilon\ \Re$ such that $x \underline{\simeq}_{\mathscr{B},\varepsilon} y$ (see Table 3). If a perceptual system is understood, then we say shortly that a set $X$ *is perceptually near to* a set $Y$ in a weak tolerance sense of *nearness*.

**Example 6. Sample Weak Tolerance Nearness**
Let $\langle O_1,\mathbb{F}\rangle$ denote perceptual system Sys. 1 with $O_1 = \{x_1, \ldots, x_9\}, \mathbb{F} = \{\phi_1,\phi_2,\phi_3,\phi_4\}$, where the values of probe functions from $\mathbb{F}$ are given in the lefthand side of table 2. Similarly, let $\langle O_2,\mathbb{F}\rangle$ denote perceptual system Sys. 2 with $O_2 = \{y_1, \ldots, y_{14}\}$, $\mathbb{F} = \{\phi_1,\phi_2,\phi_3,\phi_4\}$, where the values of the probe functions from $\mathbb{F}$ are given in the righthand side of table 2. Now choose $\varepsilon$ and arbitrary samples $X_1$ and $Y_1$ so that they are also weak tolerance near sets.

$$\begin{aligned}
&\varepsilon = 0.1,\\
&\mathscr{B} = \{\phi_3\}, \text{ where } \phi_3 \in \mathbb{F},\\
&X_1 \in O_1, Y_1 \in O_2,\\
&X_1 = \{x_1,x_2,x_7,x_8,x_9,x_{10}\},\\
&Y_1 = \{y_4,y_5,y_6,y_8,y_9,y_{11}\},\\
&X_1 \underline{\underline{\bowtie}}_{\phi_3} Y_1, \text{ since we can find } x \in X, y \in Y \text{ where } x \underline{\simeq}_{\phi_3,\varepsilon} y, \textit{ e.g.},\\
|\phi_3(x_9) - \phi_3(y_8)| &= |0.5 - 0.5| = 0 \leq 0.1; \text{ again, } \textit{e.g.},\\
|\phi_3(x_{10}) - \phi_3(y_{11})| &= |0.1 - 0.2| = 0.1
\end{aligned}$$

**Remark 3.** *In Example 6, we know that $X \underline{\underline{\bowtie}}_{\mathbb{F}} Y$, since there exists an $x \in X, y \in Y$ (namely, $x_9, y_8$) such that*

$$|\phi_3(x) - \phi_3(y)| \leq \ \varepsilon$$

*We can generalize the result from Example 6 in Prop 1 by extending the idea in Prop. 6.*

**Proposition 1.** *Let $Sys1 = \langle O_1,\mathbb{F}\rangle$ denote perceptual system Sys1. Similarly, let $Sys2 = \langle O_2,\mathbb{F}\rangle$ denote a second, distinct perceptual system. Then*

$$Sys1 \underline{\underline{\bowtie}}_{\mathbb{F}} Sys1 \iff O_1 \underline{\underline{\bowtie}}_{\mathbb{F}} O_2.$$

### 4.1 Tolerance Perceptual Near Sets

Object recognition problems, especially in images [5], and the problem of the nearness of objects have motivated the introduction of near sets (see, *e.g.*, [28]). Since we are mainly interested in real-valued probe functions in comparing swarm behaviours, perceptual near sets are briefly considered in this section based on the weak tolerance nearness relation [30] $\underline{\underline{\bowtie}}_{\mathbb{F}}$ in Def. 9. Other forms of near sets are introduced in [27, 34].

**Definition 10. Tolerance Perceptual Near Sets**
*Let* $\langle O, \mathbb{F} \rangle$ *be a perceptual system and let* $X \subseteq O$*. A set* $X$ *is a tolerance perceptual near set iff there is* $Y \subseteq O$ *such that* $X \; \underline{\underline{\bowtie}}_{\mathbb{F}} \; Y$*. The family of near sets of a perceptual system* $\langle O, \mathbb{F} \rangle$ *is denoted by* $\mathsf{Near}_{\mathbb{F}}(O)$*.*

In effect, tolerance perceptual near sets are those sets that are defined by the nearness relation $\underline{\underline{\bowtie}}_{\mathbb{F}}$.

**Example 7. Sample Tolerance Perceptual Near Sets**
Let $\langle O_1, \mathbb{F} \rangle$ denote perceptual system Sys. 1 with $O_1 = \{x_1, \ldots, x_9\}, \mathbb{F} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of probe functions from $\mathbb{F}$ are given in the lefthand side of table 2. Similarly, let $\langle O_2, \mathbb{F} \rangle$ denote perceptual system Sys. 2 with $O_2 = \{y_1, \ldots, y_{14}\}$, $\mathbb{F} = \{\phi_1, \phi_2, \phi_3, \phi_4\}$, where the values of the probe functions from $\mathbb{F}$ are given in the righthand side of table 2. Now choose samples $X$ and $Y$ that are also weak tolerance near sets. Sets $X, Y$ in Example 6 are near sets, since $X \; \underline{\underline{\bowtie}}_{\phi_3} \; Y$. Again, for example, consider the following near sets extracted from Table 2.

$$\begin{aligned}
&\varepsilon = 0.3,\\
&\mathscr{B} = \{\phi_3\},\\
&X_1 \in O_1, Y_1 \in O_2,\\
&X_1 = \{x_1, x_2, x_5, x_6, x_7, x_8, x_9, x_{10}\},\\
&Y_1 = \{y_4, y_5, y_6, y_8, y_9, y_{10}, y_{11}, y_{12}\},\\
&X_1 \; \underline{\underline{\bowtie}}_{\phi_3} Y_1, \text{ since we can find } x \in X_1, y \in Y_1, \text{ where}
\end{aligned}$$

$$x \simeq_{\phi_{3,0.3}} y, \; e.g., x_9 \simeq_{\phi_{3,0.3}} y_{10}, \text{ since } |\phi_3(x_9) - \phi_3(y_{10})| = |0.5 - 0.7| = 0.2 \leq 0.3$$

The basic idea here is to look for sets of objects containing at least one pair of objects that satisfy the weak tolerance relation. Consider, for example, sets $X_2 \in O_2, Y_1 \in O_2$ extracted from Table 2 in (23) and (24).

$$\varepsilon = 0.3 \tag{20}$$

$$\mathscr{B} = \{\phi_4\}, \tag{21}$$

$$X_2 \in O_2, Y_1 \in O_2, \tag{22}$$

$$X_2 = \{x_1, x_2, x_5, x_6, x_7, x_8, x_9\}, \tag{23}$$

$$Y_2 = \{y_5, y_6, y_8, y_9, y_{10}, y_{11}, y_{12}, y_{14}\}, \tag{24}$$

$$X_2 \; \underline{\underline{\bowtie}}_{\phi_3} Y_2, \text{ since we can find } x \in X_2, y \in Y_2, \text{ where} \tag{25}$$

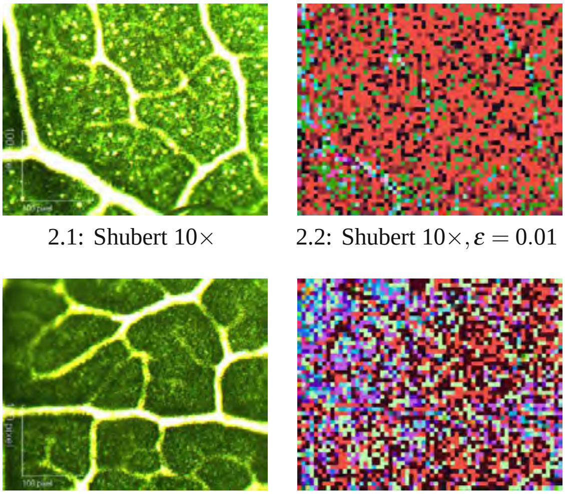
$x \cong_{\phi_{4,0.3}} y$, *e.g.*,

$x_1 \cong_{\phi_{4,0.3}} y_8$, since $|\phi_4(x_1) - \phi_4(y_8)| = |0.75 - 0.6| = 0.15 \leq 0.3$; again, *e.g.*,

$x_7 \cong_{\phi_{4,0.3}} y_{11}$, since $|\phi_4(x_7) - \phi_4(y_{11})| = |0.9 - 0.8| = 0.1 \leq 0.3$

## 4.2 Sample Near Images

By way of an illustration of near images, let $\langle Im, H \rangle$ denote a perceptual system where $Im$ is a set of segmented microscope images and $H$ is a probe function representing image entropy[2], respectively. A sample Shubert choke cherry leaf and Native Pin choke cherry leaf are shown in Figures 1.2 and 1.3. For small segments of two sample choke cherry leaves, the National Optical DC3-163 microscope in Fig. 1.1 was used to produce the magnified images in Figures 2, 3 and 4. For this example, it was found that $\gamma = 0.4239$ worked best to show the contrast between areas of the leaf fragment at the $10\times$ level of magnification in Fig. 2.1 and Fig. 2.3. Higher values of $\gamma$ were used higher levels of magnification ($\gamma = 0.874$ for $20\times$ magnification and $\gamma = 0.819$ for $40\times$ magnification).



2.1: Shubert $10\times$

2.2: Shubert $10\times, \varepsilon = 0.01$

2.3: Pin $10\times$

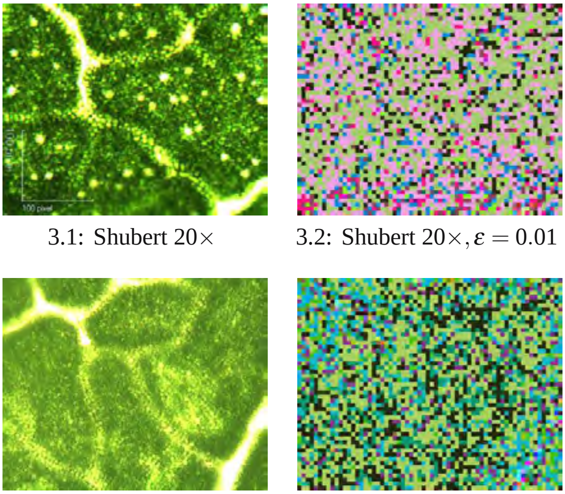
2.4: Pin $10\times, \varepsilon = 0.01$

**Fig. 2.** Sample Segmented $10\times$ Images

Let im1, im2 denote the Shubert choke cherry leaf image in Fig. 2.1 and Native pin choke cherry leaf in Fig. 2.3, respectively, each shown at magnification 10. The segmentation of these images obtained by separating image areas[3] representing tolerance classes are shown in Fig. 2.2 and Fig. 2.4. Let $\varepsilon = 0.01$ in the definition of the weak

[2] Entropy defined in the context of images is explained in [12].

[3] Christopher Henry wrote the matlab program used to obtain the image segmentations shown in this section.



3.1: Shubert 20×

3.2: Shubert $20\times, \varepsilon = 0.01$

3.3: Pin 20×

3.4: Pin $20\times, \varepsilon = 0.01$

**Fig. 3.** Sample Segmented 20× Images

tolerance relation (see Def. 4). Let $X$ denote a greyscale image, $x \in X$ a sequence of grey levels in $X$. In addition, let $p(x_i)$ denotes the probability of the $i^{th}$ sequence of grey levels). For greyscale image $X$, Pal [12] entropy $H^{(1)}$ is defined by

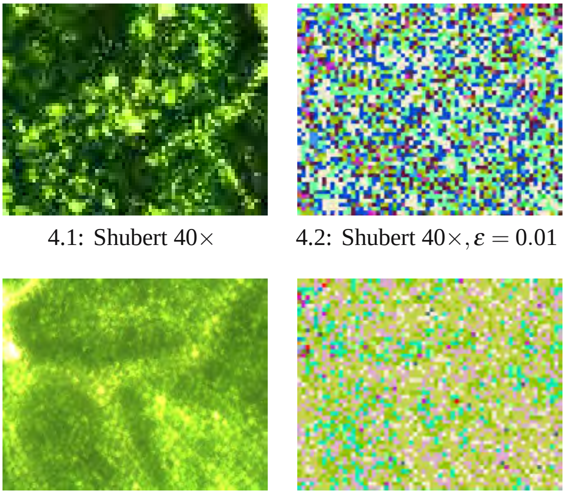
$$H(X) = \sum_{i=0}^{|X|} p(x_i) e^{1-p(x_i)}.$$

Intuitively, $H(X)$ represents the expected value of the gain in information resulting from the occurrence of different sequences of grey levels in an image. Let $x, y$ denote a pair of $n \times n$ pixel windows in an image, *i.e.*, each pixel window contains $n \times n$ pixels (picture elements). Then all pairs of pixel windows having Pal entropy within $\varepsilon = 0.01$ belong to the same tolerance class. In other words,

$$|H(x) - H(y)| \leq \varepsilon.$$

The tolerance classes represented in a segmented image are each assigned a different color. For example, the Shubert choke cherry 10× microscopic image in Fig. 2.1 is dominated by one tolerance class (visualized with tiny rectangles with the colour orange in Fig. 2.2). It can be observed that a small number of pixels windows in have the same colour. Notice that the windows in a single tolerance class are scattered throughout the image in Fig. 2.2.

A Native Pin choke cherry 10× microscopic image is shown in Fig. 2.3. The entropic pixel window values represented by the tiny rectangular regions in Fig. 2.4 are compared with the information gain (entropic image value) for each of the pixel windows shown in Fig. 2.4. For this pair of sample segmented images, roughly 20% of the pixel windows in the 10× Pin cherry segmentation have a colour (*i.e.*, information gain) that is similar to the colouring of the pixel windows in Fig. 2.2. That is, the



4.1: Shubert 40× 4.2: Shubert $40\times, \varepsilon = 0.01$

4.3: Pin 40× 4.4: Pin $40\times, \varepsilon = 0.01$

**Fig. 4.** Sample Segmented 40× Images

degree-of-nearness of this pair of images is approximately 20 percent. From Def. 4, we can conclude that $im1$ and $im2$ are near images relative to the entropic image function $H$ and for $\varepsilon = 0.01$, *i.e.*,

$$im1 \;\underline{\underline{\bowtie}}_{\{H\}}\; im2.$$

Similar results were obtained for 20× and 40× magnification levels for the segmentations shown in Fig. 3 and Fig. 4.

## 5 Rough Near Sets

The germ of the idea for near sets first appeared within a poem by Zdzisław Pawlak and this author in a poem entitled *Near To* written in 2002 and later published in English and Polish [21, 31]. In later years, the foundations for near sets grew out of a rough set approach to classifying images [5, 28, 29, 32]. It is fairly easy to show that every rough set is also a near set. This section briefly presents some fundamental notions in rough set theory resulting from the seminal work by Zdisław Pawlak during the early 1980s [20] and elaborated in [22, 23, 24]. An overview of the mathematical foundations of rough sets is given by Lech Polkowski in [35].

Let $\langle O, \mathbb{F} \rangle$ denote a perceptual system containing a set of perceptual objects $O$ and a set of functions $\mathbb{F}$ representing features of the objects in $O$. Further, let $O_{\sim_{\mathscr{B}}}$ denote the set of all classes in the partition of $O$ defined by $\sim_{\mathscr{B}}$ for $\mathscr{B} \subseteq \mathbb{F}$. Recall that $x_{/\sim_{\mathscr{B}}}$ denotes an equivalence class relative $x \in O$. For $X \subseteq O, \mathscr{B} \subseteq \mathbb{F}$, a sample perceptual granule $X$ can be approximated with a $\mathscr{B}$-lower $\mathscr{B}_*X$ and $\mathscr{B}$-upper approximation $\mathscr{B}^*X$ defined by

$$\mathscr{B}_*X = \bigcup_{x:[x]_B \subseteq X} [x]_B,$$

$$\mathscr{B}^*X = \bigcup_{x:[x]_B \cap X \neq \emptyset} [x]_B.$$

Whenever $\mathscr{B}_*X$ is a proper subset of $\mathscr{B}^*X$, *i.e.*, $\mathscr{B}^*X - \mathscr{B}_*X \neq \emptyset$, the sample $X$ has been classified imperfectly and $X$ is considered a rough set. Notice, from Def. 7,

$$\mathscr{B}_*X \bowtie_{\mathscr{B}} X, \text{ and}$$
$$\mathscr{B}^*X \bowtie_{\mathscr{B}} X,$$

since the classes in an approximation of $X$ contain objects with descriptions that match the description of at least one object in $X$. Hence, the pairs $\mathscr{B}_*X, X$ and $\mathscr{B}^*X, X$ are examples of near sets. In general,

**Proposition 2.** (Peters [27]) The pairs $(\mathscr{B}_*X, X)$ and $(\mathscr{B}^*X, X)$ are near sets.

**Proposition 3.** (Peters [27]) Any equivalence class $x_{/\sim_{\mathscr{B}}}, \left|x_{/\sim_{\mathscr{B}}}\right| > 2$ is a near set.

## 6 Fuzzy Near Sets

Fuzzy sets A1 and A2 shown in Fig. 5 are also near sets inasmuch as each fuzzy set has a non-empty core. Let $X$ be a problem domain for a fuzzy set $A$. By definition [25], the core of a fuzzy set $A_\mu$ is a function defined relative to complete and full membership in the set $A_\mu$ prescribed by the membership function $\mu$ [36]. Specifically,

$$core(A_\mu) = \{x \in X \mid \mu(x) = 1\}.$$

The core of $A_\mu$ is an example of a probe function that defines the class

$$x_{/\simeq_{core(A_\mu)}} = \left\{y \in X \mid y \in core(A_\mu)\right\}.$$

It can also be argued that $\langle X, core(A_\mu)\rangle$ is a perceptual system. In the case where a pair of fuzzy sets has non-empty cores, then the fuzzy sets satisfy the condition for the

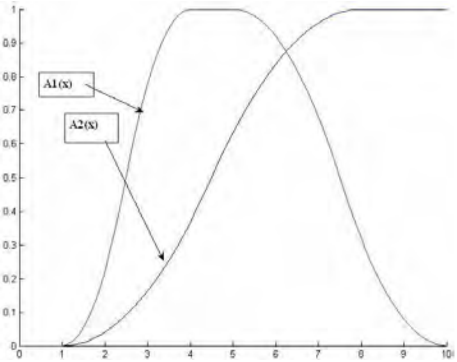


**Fig. 5.** Sample Fuzzy Near Sets

weak nearness relation, *i.e.*, we can find $x \in X, y \in Y$ for $\left(X, A1_{\mu_1}\right), \left(Y, A2_{\mu_2}\right)$ relative to $A1_{\mu_1}, A2_{\mu_2}$, for membership functions $\mu_1, \mu_2$, where

$$x \in x_{/\simeq_{core(A1_{\mu_1})}},$$
$$y \in y_{/\simeq_{core(A2_{\mu_2})}},$$
$$\mu_1(x) = \mu_2(y) = 1.$$

**Proposition 4.** Fuzzy sets with non-empty cores are near sets.

## 7 Conclusion

The main contribution of this chapter is the introduction of a formal basis for discovering affinities between perceptual granules. This is made possible by the introduction of various forms of indiscernibility relations that define partitions and tolerance relations that define coverings of perceptual granules and lead to a number of useful nearness relations. A weak tolerance nearness relation is also defined in this chapter. This tolerance relation is has proven to be quite useful in discovering affinities between perceptual granules. The degree of affinity between microscope images as a perceptual system is measured with a form of entropic image function, has been briefly presented in an informal way in this chapter. Future work includes the introduction of various probe functions and nearness useful in image analysis.

**Acknowledgements.** The comments, suggestions and questions by anonymous reviewer for this article were very helpful. Many thanks to Christopher Henry for providing the MatLab code for the image segmentations presented in this chapter. This research has been funded by grant SRI-BIO-05 from the Canadian Arthritis Network (CAN) and by discovery grants 185986 and 194376 from the Natural Sciences & Engineering Research Council of Canada (NSERC).

## References

1. Bargiela, A., Pedrycz, W.: Granular Computing. Kluwer, Boston (2003)
2. Calitoiu, D., Oommen, B., Nussbaum, D.: Desynchronizing a chaotic pattern recognition neural network to model inaccurate perception. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics 37(3), 692–704 (2007)
3. Darwin, C.: On the Origin of the Species by Means of Natural Selection. J. Murray, Oxford (1859)
4. Fahle, M., Poggio, T.: Perceptual Learning. The MIT Press, Cambridge (2002)
5. Henry, C., Peters, J.F.: Image pattern recognition using approximation spaces and near sets. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) RSFDGrC 2007. LNCS, vol. 4482, pp. 475–482. Springer, Heidelberg (2007)
6. Jähne, B.: Digital Image Processing, 6th edn. Springer, Heidelberg (2005)
7. Merleau-Ponty, M.: Phenomenology of Perception, trans. by Colin Smith. Smith, Callimard and Routledge & Kegan Paul, Paris and New York (1945, 1956)

8. Murray, J., Bradley, H., Craigie, W., Onions, C.: The Oxford English Dictionary. Oxford University Press, Oxford (1933)
9. Nasrabadi, N., Pal, S., King, R.: Entropy-coded hybrid differential pulse-code modulation. Electronics Letters 16(2), 63–65 (1983)
10. Orłowska, E.: Incomplete Information: Rough Set Analysis. Studies in Fuzziness and Soft Computing, vol. 13. Physica-Verlag, Heidelberg (1998)
11. Orłowska, E., Pawlak, Z.: Representation of nondeterministic information. Theoretical Computer Science 29, 27–39 (1984)
12. Pal, N.R., Pal, S.K.: Entropy: A new definition and its applications. IEEE Trans. on Sys., Man and Cybernetics 21(5), 1260–1270 (1981)
13. Pal, N.R., Pal, S.K.: Entropic thresholding. Signal Processing 16(2), 97–108 (1983)
14. Pal, N.R., Pal, S.K.: Object-background segmentation using new definitions of entropy. IEE Proceedings-E 136(4), 284–295 (1989)
15. Pal, N.R., Pal, S.K.: Entropy: A new definition and its applications. IEEE Trans. Syst, Man and Cybernetics SMC-21(5), 1260–1270 (1991)
16. Pal, N.R., Pal, S.K.: Higher order fuzzy entropy and hybrid entropy of a set. Information Sciences 61(3), 211–231 (1992)
17. Pal, N.R., Pal, S.K.: Some properties of the exponential entropy. Information Sciences 66(3), 119–137 (1992)
18. Pal, S.K., King, R.A., Hashim, A.A.: Automatic gray level thresholding through index of fuzziness and entropy. Pattern Recognition Letters 1, 141–146 (1983)
19. Pavel, M.: Fundamentals of Pattern Recognition, 2nd edn. Marcel Dekker, Inc., N.Y (1993)
20. Pawlak, Z.: Classification of objects by means of attributes. Polish Academy of Sciences 429 (1981)
21. Pawlak, Z., Peters, J.F.: Jak blisko. Systemy Wspomagania Decyzji I, 57 (2007)
22. Pawlak, Z., Skowron, A.: Rough sets and boolean reasoning. Information Sciences 177, 41–73 (2007)
23. Pawlak, Z., Skowron, A.: Rough sets: Some extensions. Information Sciences 177, 28–40 (2007)
24. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Information Sciences 177, 3–27 (2007)
25. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. The MIT Press, Cambridge (1998)
26. Peters, J.F.: Classification of objects by means of features. In: Proc. IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SCCI 2007), Honolulu, Hawaii, pp. 1–8 (2007)
27. Peters, J.: Near sets. general theory about nearness of objects. Applied Mathematical Sciences 1(53), 2609–2629 (2007)
28. Peters, J.F.: Near sets. special theory about nearness of objects. Fundamenta Informaticae 75(1-4), 407–433 (2007)
29. Peters, J.F.: Near sets. Toward approximation space-based object recognition. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 22–33. Springer, Heidelberg (2007)
30. Peters, J.: Discovery of perceputally near information granules. In: Novel Developments in Granular Computing: Applications of Advanced Human Reasoning and Soft Computation. Information Science Reference, Hersey, N.Y., U.S.A (to appear, 2008)
31. Peters, J.F., Skowron, A.: Zdzisław pawlak: Life and work. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets V. LNCS, vol. 4100, pp. 1–25. Springer, Heidelberg (2006)
32. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness in approximation spaces. In: Proc. Concurrency, Specification and Programming (CS&P 2006), Humboldt Universität, pp. 435–445 (2006)

33. Peters, J.F., Skowron, A., Stepaniuk, J.: Nearness of objects: Extension of approximation space model. Fundamenta Informaticae 79(3-4), 497–512 (2007)
34. Peters, J.F., Wasilewski, P.: Foundations of near sets. Information Sciences (submitted, 2008)
35. Polkowski, L.: Rough Sets. Mathematical FoundationsGermany. Springer, Heidelberg (2002)
36. Ross, T.J.: Fuzzy Logic with Engineering Applications. John Wiley & Sons, New York (2004)
37. Zeeman, E.: The topology of the brain and visual perception. In: Fort Jr., M.K. (ed.) Topology of 3-Manifolds and Related Topics, University of Georgia Institute, pp. 240–256. Prentice-Hall, Inc., Englewood Cliffs (1961)

# Concept Granular Computing Based on Lattice Theoretic Setting

Wen-xiu Zhang[1], Hong-zhi Yang[1], Jian-min Ma[2], and Guo-fang Qiu[3]

[1] Faculty of Science, Xi'an Jiaotong University, Shaanxi 710049, P.R.China
wxzhang@mail.xjtu.edu.cn, liuyepd@163.com

[2] Department of Mathematics and Information Science, Faculty of Science,
Chang'an University, Shaanxi 710064, P.R.China
cjm-zm@126.com

[3] School of Management, Xi'an University of Architecture & Technology,
Shaanxi 710055, P.R.China
qiugf_home@163.com

**Abstract.** Based on the theory of concept lattice and fuzzy concept lattice, a mathematical model of a concept granular computing system is established, and relationships of the system and concept lattices, various variable threshold concept lattices and fuzzy concept lattices are then investigated. For this system, concept granules, sufficiency information granules and necessity information granules which are used to express different relations between a set of objects and a set of attributes are proposed. Approaches to construct sufficiency and necessity information granules are also shown. Some iterative algorithms to form concept granules are proposed. It is proved that the concept granules obtained by the iterative algorithms are the sub-concept granules or sup-concept granules under some conditions for this system. Finally, we give rough approximations based on fuzzy concept lattice in formal concept analysis.

## 1 Introduction

A concept is the achievement of human recognizing the world. It announces the essence to distinguish one object from the others. Meanwhile, a concept is also a unit of human thinking and reasoning. New concepts are often produced by the original known ones. Thus, a concept is regarded as an information granule, and it plays an important role in our perception and recognition. In 1979, Zadeh first introduced the notion of fuzzy information granules [45]. From then on, many researchers paid much attention to the thought of information granules, and applied it to many fields such as rough set, fuzzy set and evidence reasoning [14,19]. The notion of granularity was proposed by Hobbs in 1985 [20], and granular computing was first provided by Zadeh from 1996 to 1997 [46,47]. Since then, granular computing as a fundamental thought has stepped to soft computing, knowledge discovery and data mining, and has obtained some good results [26,32,41].

Formal concept analysis (FCA), proposed by Wille in 1982 [37], is a mathematical framework for discovery and design of concept hierarchies from a formal context. It is an embranchment of applied mathematics, which made it need mathematical thinking

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 67–94.
springerlink.com

for applying FCA to data analysis and knowledge processing [13]. All formal concepts of a formal context with their specification and generalization form a concept lattice [18]. And the concept lattice can be depicted by a Hassen diagram, where each node expresses a formal concept. The concept lattice is the core structure of data in FCA. In essence, a formal concept represents a relationship between the extension of a set of objects and the intension of a set of attributes, and the extension and the intension are uniquely determined each other. The more the formal concepts can be obtained, the stronger the ability is to recognize the world. Thus FCA is regarded as a power tool for learning problems [11,21,22,24,25].

Recently, there has been much advance in the study for FCA, especially in the study of the combination of FCA with the theory of rough set [12,23,28,30,35,38]. Zhang etc. proposed the theory and approach of attribute reduction of concept lattice with the formal context being regarded as a 0-1 information table, and introduced the judgment theorems of attribute reduction [48-50]. In their paper, they also introduced a decision formal context, and then acquired decision rules from it. Yao studied relations between FCA and the theory of rough set [42-44]. Burusco and Belohlavek investigated fuzzy concept lattices of *L*-fuzzy formal context [1-10]. Fan etc. discussed reasoning algorithm of the fuzzy concept lattice based on a complete residuated lattice, studied the relationships among various variable threshold fuzzy concept lattices, and proposed fuzzy inference methods [17]. Ma etc. constructed relations between fuzzy concept lattices and granular computing [26]. Qiu gave the iterative algorithms of concept lattices [29]. Shao etc. established the set approximation in FCA [31].

In this paper, a mathematical model of a concept granular computing system is introduced based on the study of concept lattice and fuzzy concept lattice. Relationships among this system and concept lattice, fuzzy concept lattice and variable threshold concept lattice are investigated. Properties of the system are then studied. To describe the relations between a set of objects and a set of attributes, sufficiency information granules and necessity information granules are defined. Iterative algorithms of a concept granular computing system are proposed to obtain the information granules. And rough approximations of a set based on the concept lattice are studied. It may supply another way to study FCA.

This paper is organized as follows. In section 2, we review basic notions and properties of concept lattice and fuzzy concept lattice. Then we propose a mathematical model called a concept granular computing system in Section 3. Relationships among this system and concept lattice, variable threshold concept lattice and fuzzy concept lattice are investigated. In Section 4, we study properties of this system. And sufficiency information granules and necessity information granules are defined in Section 5. We propose some iterative algorithms to produce concept granules in Section 6. Finally, set approximation in FCA is studied in Section 7. The paper is then concluded with a summary in Section 8.

## 2 Preliminaries

To facilitate our discussion, this section reviews some notions and results related to concept lattice and fuzzy concept lattice. The following definitions and theorems are the relevant facts about concept lattice and fuzzy concept lattice [2,18,19].

In FCA, the data for analysis is described as a formal context, on which we can construct formal concepts. All formal concepts form a concept lattice which explains hierarchical relations of concepts.

**Definition 1.** A triplet $(U, A, I)$ is called a formal context, where $U = \{x_1, \cdots, x_n\}$ is a nonempty and finite set called the universe of discourse, every element $x_i (i \le n)$ is an object; $A = \{a_1, \cdots, a_m\}$ is a nonempty and finite set of attributes, every element $a_j (j \le m)$ is an attribute; and $I \subseteq U \times A$ is a binary relation between $U$ and $A$.

For a formal context $(U, A, I)$, $x \in U$ and $a \in A$, we use $(x, a) \in I$, or $xIa$, denotes that the object $x$ has the attribute $a$. If we use 1 and 0 to express $(x, a) \in I$ and $(x, a) \notin I$, respectively, then the formal context can be described as a 0-1 information table.

Let $(U, A, I)$ be a formal context, $X \subseteq U$ and $B \subseteq A$, we define a pair of operators:

$$X^* = \{a : a \in A, \forall x \in X, xIa\} \tag{1}$$

$$B^* = \{x : x \in U, \forall a \in B, xIa\} \tag{2}$$

where $X^*$ denotes the set of attributes common to the objects in $X$, and $B^*$ is the set of objects possessing all attributes in $B$. For simplicity, for any $x \in U$ and $a \in A$, we use $x^*$ and $a^*$ instead of $\{x\}^*$ and $\{a\}^*$, respectively. For any $x \in U$ and $a \in A$, if $x^* \ne \varnothing$, $x^* \ne A$, and $a^* \ne \varnothing$, $a^* \ne U$, we call the formal context $(U, A, I)$ is regular. In this paper, we suppose the formal contexts we discussed are regular.

**Definition 2.** Let $(U, A, I)$ be a formal context, $X \subseteq U$ and $B \subseteq A$. A pair $(X, B)$ is referred to as a formal concept, or a concept if $X^* = B$ and $X = B^*$. We call $X$ the extension and $B$ the intension of the concept $(X, B)$.

**Proposition 1.** Let $(U, A, I)$ be a formal context. Then for any $X_1, X_2, X \subseteq U$ and $B_1, B_2, B \subseteq A$, we can obtain that:

(P1) $X_1 \subseteq X_2 \Rightarrow X_2^* \subseteq X_1^*$, $B_1 \subseteq B_2 \Rightarrow B_2^* \subseteq B_1^*$;

(P2) $X \subseteq X^{**}$, $B \subseteq B^{**}$;

(P3) $X^* = X^{***}$, $B^* = B^{***}$;

(P4) $X \subseteq B^* \Leftrightarrow B \subseteq X^*$;

(P5) $(X_1 \cup X_2)^* = X_1^* \cap X_2^*$, $(B_1 \cup B_2)^* = B_1^* \cap B_2^*$;

(P6) $(X_1 \cap X_2)^* \supseteq X_1^* \cup X_2^*$, $(B_1 \cap B_2)^* \supseteq B_1^* \cup B_2^*$;

(P7) $(X^{**}, X^*)$ and $(B^*, B^{**})$ are always concepts.

In Definition 2, concepts are constructed based on a classical formal context with the binary relation between objects and attributes being either 0 or 1. In the real world, however, the binary relation between objects and attributes are fuzzy and uncertain. Burusco etc. extended the classical model to a fuzzy formal context [8,40], on which fuzzy concepts are first established.

Let $L$ be a complete lattice. We denote by $L^U$ the set of all $L$-fuzzy sets defined on $U$. Then for any $L$-fuzzy sets $\tilde{X}_1, \tilde{X}_2 \in L^U$, for any $x \in U$, $\tilde{X}_1 \subseteq \tilde{X}_2 \Leftrightarrow \tilde{X}_1(x) \le \tilde{X}_2(x)$. Then $(L^U, \subseteq)$ forms a poset. Obviously, $([0,1]^U, \subseteq)$ and $(\{0,1\}^U, \subseteq)$ are both posets.

We denote by $P(U)$ and $P(A)$ the power set on the universe of discourse $U$ and the power set on the set of attributes $A$, respectively.

**Definition 3.** A triplet $(U, A, \tilde{I})$ is referred to as a $L$-fuzzy formal context, where $U$ is a universe of discourse, $A$ is a nonempty and finite set of attributes, and $\tilde{I}$ is a $L$-fuzzy relation between $U$ and $A$, i.e. $\tilde{I} \in L^{U \times A}$.

$\boldsymbol{L} = (L, \vee, \wedge, \otimes, \to, 0, 1)$ is referred to as a complete residuated lattice, if $(L, \vee, \wedge, 0, 1)$ is a complete lattice with the least element 0 and the great element 1; $(L, \otimes, 1)$ is a commutative semigroup with unit element 1; and $(\otimes, \to)$ is a residuated pair of $L$, i.e. $\otimes : L \times L \to L$ is monotone increasing , $\to: L \times L \to L$ is non-increasing for the first variable and non-decreasing for the second variable, and for any $a, b, c \in L, a \otimes b \le c \Leftrightarrow a \le b \to c$.

Let $(U, A, \tilde{I})$ be a $L$-fuzzy formal context, $\tilde{X} \in L^U$ and $\tilde{B} \in L^A$. We define two operators as follows:

$$\tilde{X}^+(a) = \wedge_{x \in U} (\tilde{X}(x) \to \tilde{I}(x, a)) \tag{3}$$

$$\tilde{B}^+(x) = \wedge_{a \in A} (\tilde{B}(a) \to \tilde{I}(x, a)) \tag{4}$$

Then $\tilde{X}^+ \in L^A$ and $\tilde{B}^+ \in L^U$.

**Definition 4.** Let $(U, A, \tilde{I})$ be a $L$-fuzzy formal context. $(\tilde{X}, \tilde{B})$ is referred to as a fuzzy formal concept, or a fuzzy concept if $\tilde{X}^+ = \tilde{B}$ and $\tilde{B}^+ = \tilde{X}$ for any $\tilde{X} \in L^U$ and $\tilde{B} \in L^A$.

**Proposition 2.** Let $(U, A, \tilde{I})$ be a $L$-fuzzy formal context, $\boldsymbol{L} = (L, \vee, \wedge, \otimes, 0, 1)$ be a complete residuated lattice. Then for any $\tilde{X}_1, \tilde{X}_2, \tilde{X} \in L^U$ and $\tilde{B}_1, \tilde{B}_2, \tilde{B} \in L^A$, we have the following properties:

(F1) $\tilde{X}_1 \subseteq \tilde{X}_2 \Rightarrow \tilde{X}_2^+ \subseteq \tilde{X}_1^+$, $\tilde{B}_1 \subseteq \tilde{B}_2 \Rightarrow \tilde{B}_2^+ \subseteq \tilde{B}_1^+$;

(F2) $\tilde{X} \subseteq \tilde{X}^{++}$, $\tilde{B} \subseteq \tilde{B}^{++}$;

(F3) $\tilde{X}^+ = \tilde{X}^{+++}$, $\tilde{B}^+ = \tilde{B}^{+++}$;

(F4) $\tilde{X} \subseteq \tilde{B}^+ \Leftrightarrow \tilde{B} \subseteq \tilde{X}^+$;

(F5) $(\tilde{X}_1 \bigcup \tilde{X}_2)^+ = \tilde{X}_1^+ \bigcap \tilde{X}_2^+$, $(\tilde{B}_1 \bigcup \tilde{B}_2)^+ = \tilde{B}_1^+ \bigcap \tilde{B}_2^+$;

(F6) $(\tilde{X}^{++}, \tilde{X}^+)$ and $(\tilde{B}^+, \tilde{B}^{++})$ are always fuzzy concepts.

**Proposition 3.** Let $(U, A, \tilde{I})$ be a *L*-fuzzy formal context. Note that

$$L_f(U, A, \tilde{I}) = \{(\tilde{X}, \tilde{B}) : \tilde{X}^+ = \tilde{B}, \tilde{B}^+ = \tilde{X}\}$$

For any $(\tilde{X}_1, \tilde{B}_1)$, $(\tilde{X}_2, \tilde{B}_2) \in L_f(U, A, \tilde{I})$, we define a binary relation $"\leq"$ as follows:

$$(\tilde{X}_1, \tilde{B}_1) \leq (\tilde{X}_2, \tilde{B}_2) \Leftrightarrow \tilde{X}_1 \subseteq \tilde{X}_2 ( \Leftrightarrow \tilde{B}_1 \supseteq \tilde{B}_2 ).$$

Then $"\leq"$ is a partial order on $L_f(U, A, \tilde{I})$, and $(L_f(U, A, \tilde{I}), \leq)$ is a complete lattice, called fuzzy concept lattice, in which the meet and join operators are given by:

$$\underset{i\in T}{\wedge}(\tilde{X}_i, \tilde{B}_i) = (\underset{i\in T}{\cap} \tilde{X}_i, (\underset{i\in T}{\cup} \tilde{B}_i)^{++}),$$
$$\underset{i\in T}{\vee}(\tilde{X}_i, \tilde{B}_i) = ((\underset{i\in T}{\cup} \tilde{X}_i)^{++}, \underset{i\in T}{\cap} \tilde{B}_i).$$

where $T$ is a finite index set.

Obviously, a classical formal context is a special *L*-fuzzy formal context, i.e. formula (1) and (2) are special situations of formula (3) and (4), respectively.

## 3 Mathematical Model of Concept Granular Computing System

For a formal context and a fuzzy formal context, by constructing operators between the set of objects and the set of attributes, we obtain concept lattice and fuzzy concept lattice. In this section, we extend the formal context to a generalized setting, and then obtain a mathematical model for concept granular computing system.

Let $L$ be a complete lattice. We denote by $0_L$ and $1_L$ the zero element and the unit element of $L$, respectively.

**Definition 5.** Let $L_1, L_2$ be two complete lattices. We call any element in $L_1$ an extent element and any elements in $L_2$ an intent element. The mapping $G : L_1 \to L_2$ is referred to as an extent-intent operator if it satisfies:

(G1) $G(0_{L_1}) = 1_{L_2}, G(1_{L_1}) = 0_{L_2}$;

(G2) $G(a_1 \vee a_2) = G(a_1) \wedge G(a_2), \forall a_1, a_2 \in L_1$.

For any $a \in L_1$ , $G(a)$ is called the intent element of $a$ . The mapping $H: L_2 \to L_1$ is referred to as an intent-extent operator if it satisfies:

(H1) $H(0_{L_2}) = 1_{L_1}, H(1_{L_2}) = 0_{L_1}$;

(H2) $H(b_1 \vee b_2) = H(b_1) \wedge H(b_2), \forall b_1, b_2 \in L_2$.

For any $b \in L_2$ , $H(b)$ is called the extent element of $b$ .

**Definition 6.** Let $G$ and $H$ be the extent-intent and intent-extent operators on $L_1$ and $L_2$, respectively. Furthermore, if for any $a \in L_1$ and $b \in L_2$ ,

$$a \le H \circ G(a), b \le G \circ H(b),$$

the quadruplex $(L_1, L_2, G, H)$ is referred to as a concept granular computing system, where $H \circ G(a), G \circ H(b)$ is described as $H(G(a))$ and $G(H(b))$ respectively.

**Theorem 1.** Let $(U, A, I)$ be a formal context. Then the operators $(*,*)$ defined by formula (1) and (2) are extent-intent and intent-extent operators, respectively. And $(P(U), P(A), *, *)$ is a concept granular computing system.

*Proof.* It immediately follows from Proposition 1.

**Theorem 2.** Let $(U, A, \tilde{I})$ be a *L*-fuzzy formal context. Then the operators defined by formula (3) and (4) are extent-intent and intent-extent operators, respectively. And $(L^U, L^A, +, +)$ is a concept granular computing system.

*Proof.* It immediately follows from Proposition 2.

**Theorem 3.** Let $(P(U), P(A), G, H)$ be a concept granular computing system. Then there exists a binary relation $I \subseteq U \times A$ such that $(U, A, I)$ is a formal context, and $(*,*) = (G, H)$.

*Proof.* Note that

$$I = \{(x, a) : x \in H(\{a\})\}$$

Then $(U, A, I)$ is a formal context. For any $B \subseteq A$, we have $H(B) = \bigcap_{a \in B} H(\{a\})$ by (H2). Thus

$$\begin{aligned} B^* &= \{x \in U : \forall a \in B, x \in H(\{a\})\} \\ &= \{x \in U : x \in \bigcap_{a \in B} H(\{a\}) = H(B)\} = H(B) \end{aligned}$$

By (G2), for any $X_1 \subseteq X_2$ , we have

$$G(X_2) = G(X_1 \cup X_2) = G(X_1) \cap G(X_2).$$

Then $X_1 \subseteq X_2$ implies $G(X_2) \subseteq G(X_1)$. Thus, for $x \in H(\{a\})$, we have $G(\{x\}) \supseteq G(H(\{a\})) \supseteq \{a\}$. That is, $a \in G(\{x\})$. Analogously, we can prove that $B_1 \subseteq B_2 \Rightarrow H(B_2) \subseteq H(B_1)$. By which and $a \in G(\{x\})$ we can get that $x \in H(\{a\})$. Thus $x \in H(\{a\})$ iff $a \in G(\{x\})$. Therefore,

$$I = \{(x,a) : a \in G(\{x\})\}.$$

For any $X \subseteq U$, by property (G2) we have $G(X) = \bigcap_{x \in X} G(\{x\})$. Thus,

$$\begin{aligned} X^* &= \{a \in A : \forall x \in X, a \in G(\{x\})\} \\ &= \{a \in A : a \in \bigcap_{x \in X} G(\{x\}) = G(X)\} = G(X). \end{aligned}$$

We denote by $L = [0,1]$ a unit interval. Then $\boldsymbol{L} = ([0,1], \vee, \wedge, \otimes, \to, 0, 1)$ is a complete residuated lattice. We call the $L$-fuzzy formal context $(U, A, \tilde{I})$ with $L = [0,1]$ a fuzzy formal context. Then for any $X \in P(U)$, $B \in P(A)$ and $0 < \delta \le 1$, we define two operators as follows:

$$X^{\#} = \{a \in A : \bigwedge_{x \in X} (X(x) \to \tilde{I}(x,a)) \ge \delta\} \tag{5}$$

$$B^{\#} = \{x \in U : \bigwedge_{a \in B} (B(a) \to \tilde{I}(x,a)) \ge \delta\} \tag{6}$$

**Theorem 4.** A quadruplex $(P(U), P(A), \#, \#)$ is a concept granular computing system.

*Proof.* Obviously, $P(U), P(A)$ are complete lattices. According to formula (5), the operator $\# : P(U) \to P(A)$ satisfies $\varnothing^{\#} = A$ and $U^{\#} = \varnothing$. Since $\vee x_i \to a = \wedge (x_i \to a)$, then for any $X_1, X_2 \in P(U)$, it follows that

$$\begin{aligned} (X_1 \cup X_2)^{\#} &= \{a \in A : \bigwedge_{x \in X_1 \cup X_2} ((X_1(x) \vee X_2(x)) \to \tilde{I}(x,a)) \ge \delta\} \\ &= \{a \in A : \bigwedge_{x \in X_1 \cup X_2} ((X_1(x) \to \tilde{I}(x,a)) \wedge (X_2(x) \to \tilde{I}(x,a))) \ge \delta\} \\ &= \{a \in A : \bigwedge_{x \in X_1} (X_1(x) \to \tilde{I}(x,a)) \ge \delta\} \cap \{a \in A : \bigwedge_{x \in X_2} (X_2(x) \to \tilde{I}(x,a)) \ge \delta\} \\ &= X_1^{\#} \cap X_2^{\#}. \end{aligned}$$

Thus, the operator $\# : P(U) \to P(A)$ is an extent-intent operator. Similarly, we can prove the operator $\# : P(A) \to P(U)$ is an intent-extent operator.

Meanwhile, for any $X \in P(U)$, since

$$X^{\#} = \{a \in A : \bigwedge_{x \in X} (X(x) \to \tilde{I}(x,a)) \ge \delta\},$$
$$X^{\#\#} = \{x \in U : \bigwedge_{a \in X^{\#}} (X^{\#}(a) \to \tilde{I}(x,a)) \ge \delta\},$$

by formula (5) we have, for any $x \in X$, if $a \in X^{\#}$, we have $1 \to \tilde{I}(x,a) \geq \delta$. Because $X^{\#}$ is a crisp set, we have $X^{\#\#} = \{x \in U : \underset{a \in X^{\#}}{\wedge}(1 \to \tilde{I}(x,a)) \geq \delta\}$. Therefore, $x \in X$ implies $\underset{a \in X^{\#}}{\wedge}(X^{\#}(a) \to \tilde{I}(x,a)) \geq \delta$. That is, $x \in X^{\#\#}$. Thus, $X \subseteq X^{\#\#}$. Similarly, it can be proved that for any $B \in P(A)$, $B \subseteq B^{\#\#}$. Thus, $(P(U), P(A), \#, \#)$ is a concept granular computing system.

Let $(U, A, \tilde{I})$ be a fuzzy formal context. For any $X \in P(U)$, $\tilde{B} \in L^{A}$, and $0 < \delta \leq 1$, a pair of operators are defined as follows:

$$X^{\Delta}(a) = \delta \to \underset{x \in X}{\wedge} \tilde{I}(x,a) \qquad (a \in A) \tag{7}$$

$$\tilde{B}^{\nabla} = \{x \in U : \underset{a \in A}{\wedge}(\tilde{B}(a) \to \tilde{I}(x,a)) \geq \delta\} \tag{8}$$

**Theorem 5.** A quadruplex $(P(U), L^{A}, \Delta, \nabla)$ is a concept granular computing system.

*Proof.* It is similarly proved as Theorem 4.

Let $(U, A, \tilde{I})$ be a fuzzy formal context. For any $\tilde{X} \in L^{U}$, $B \in P(A)$, and $0 < \delta \leq 1$, a pair of operators are defined as follows:

$$\tilde{X}^{\nabla} = \{a \in A : \underset{x \in U}{\wedge}(\tilde{X}(x) \to \tilde{I}(x,a)) \geq \delta\} \tag{9}$$

$$B^{\Delta}(x) = \delta \to \underset{a \in B}{\wedge} \tilde{I}(x,a) \qquad (x \in U) \tag{10}$$

**Theorem 6.** A quadruplex $(L^{U}, P(A), \nabla, \Delta)$ is a concept granular computing system.

*Proof.* It is similarly proved as Theorem 4.

## 4 Properties of Concept Granular Computing System

**Definition 7.** Let $(L_1, L_2, G, H)$ be a concept granular computing system. If for any $a \in L_1$ and $b \in L_2$, $G(a) = b$ and $H(b) = a$, then the pair $(a,b)$ is called a concept. We call $a$ the extension and $b$ the intension of the concept $(a,b)$.

For any concepts $(a_1, b_1), (a_2, b_2)$, we define a binary relation "$\leq$"as follows:

$$(a_1, b_1) \leq (a_2, b_2) \Leftrightarrow a_1 \leq a_2 .$$

Then "$\leq$"is a partial order.

Let $(L_1, L_2, G, H)$ be a concept granular computing system. By the operators $G$ and $H$, a bridge between the extent set and the intent set is constructed, which describe the transformation process of objects and attributes for the recognition.

**Theorem 7.** Let $(L_1, L_2, G, H)$ be a concept granular computing system. Then the following conclusions hold:

(1) $a_1 \le a_2 \Rightarrow G(a_2) \le G(a_1)$, for any $a_1, a_2 \in L_1$;

(2) $b_1 \le b_2 \Rightarrow H(b_2) \le H(b_1)$, for any $b_1, b_2 \in L_2$;

(3) $G(a_1) \vee G(a_2) \le G(a_1 \wedge a_2)$ for any $a_1, a_2 \in L_1$;

(4) $H(b_1) \vee H(b_2) \le H(b_1 \wedge b_2)$, for any $b_1, b_2 \in L_2$;

(5) $G \circ H \circ G(a) = G(a)$ for any $a \in L_1$;

(6) $H \circ G \circ H(b) = H(b)$ for any $b \in L_2$.

*Proof.* (1) Suppose $a_1, a_2 \in L_1$, and $a_1 \le a_2$. Since $G$ is an extent-intent operator, we have

$$G(a_2) = G(a_1 \vee a_2) = G(a_1) \wedge G(a_2).$$

Thus, $G(a_2) \le G(a_1)$.

(2) It is similarly proved as (1).

(3) Because $a_1 \wedge a_2 \le a_1$ and $a_1 \wedge a_2 \le a_2$, by (1) we can get that

$$G(a_1) \le G(a_1 \wedge a_2) \text{ and } G(a_2) \le G(a_1 \wedge a_2).$$

Then $G(a_1) \vee G(a_2) \le G(a_1 \wedge a_2)$.

(4) It is similarly proved as (3).

(5) Since for any $a \in L_1$, $a \le H \circ G(a)$, then by (1) we can get that $G \circ H \circ G(a) \le G(a)$. Meanwhile, let $b = G(a)$, we have $b \le G \circ H(b)$. Thus, $G(a) \le G \circ H \circ G(a)$, which leads to $G(a) = G \circ H \circ G(a)$.

(6) It is similarly proved as (5).

**Theorem 8.** Let $(L_1, L_2, G, H)$ be a concept granular computing system. Note that

$$\mathfrak{B}(L_1, L_2, G, H) = \{(a,b) : G(a) = b, H(b) = a\}$$

Then $\mathfrak{B}(L_1, L_2, G, H)$ is a lattice with a great element and a least element, where the infimum and the supremum are defined as follows

$$\underset{i\in T}{\wedge}(a_i, b_i) = (\underset{i\in T}{\wedge} a_i, G \circ H(\underset{i\in T}{\vee} b_i)),$$

$$\underset{i\in T}{\vee}(a_i, b_i) = (H \circ G(\underset{i\in T}{\vee} a_i), \underset{i\in T}{\wedge} b_i),$$

where $(a_i, b_i) \in \mathfrak{B}(L_1, L_2, G, H)$ ($i \in T, T$ is a finite index set).

*Proof.* Since $(a_i, b_i) \in \mathfrak{B}(L_1, L_2, G, H)$, we have $G(a_i) = b_i, H(b_i) = a_i$. Thus,

$$G(\underset{i\in T}{\wedge} a_i) = G(\underset{i\in T}{\wedge} H(b_i)) = G(H(\underset{i\in T}{\vee} b_i)) = G \circ H(\underset{i\in T}{\vee} b_i),$$

$$H \circ G \circ H(\vee_{i\in T} b_i) = H(\vee_{i\in T} b_i) = \wedge_{i\in T} H(b_i) = \wedge_{i\in T} a_i .$$

Then $\wedge_{i\in T}(a_i, b_i) \in \mathfrak{B}(L_1, L_2, G, H)$ . Similarly, we can prove that

$$\vee_{i\in T}(a_i, b_i) \in \mathfrak{B}(L_1, L_2, G, H) .$$

Since $G(0_{L_1}) = 1_{L_2}, H(1_{L_2}) = 0_{L_1}$ , by the partial order $\leq$ we have $(0_{L_1}, 1_{L_2})$ is the least element of $\mathfrak{B}(L_1, L_2, G, H)$ . Similarly, $(1_{L_1}, 0_{L_2})$ is the great element of $\mathfrak{B}(L_1, L_2, G, H)$ .

In order to prove $\mathfrak{B}(L_1, L_2, G, H)$ is a lattice, we need to prove that $\wedge_{i\in T}(a_i, b_i)$ is the great lower bound of $(a_i, b_i)(i \in T)$ and $\vee_{i\in T}(a_i, b_i)$ is the least upper bound of $(a_i, b_i)(i \in T)$ . Since $\wedge_{i\in T} a_i \leq a_i$ , we have $\wedge_{i\in T}(a_i, b_i) = (\wedge_{i\in T} a_i, G \circ H(\vee_{i\in T} b_i)) \leq (a_i, b_i)$ . That is, $\wedge_{i\in T}(a_i, b_i)$ is the lower bound of $(a_i, b_i)(i \in T)$ . Suppose $(a, b) \in \mathfrak{B}(L_1, L_2, G, H)$ and $(a, b) \leq (a_i, b_i)$ for any $i \in T$ . Then $(a, b) \leq \wedge_{i\in T}(a_i, b_i)$ , and we can get that $\wedge_{i\in T}(a_i, b_i)$ is the great lower bound of $(a_i, b_i)(i \in T)$ . Similarly, we can prove that $\vee_{i\in T}(a_i, b_i)$ is the least upper bound of $(a_i, b_i)(i \in T)$ . Therefore, $\mathfrak{B}(L_1, L_2, G, H)$ is a lattice with a great element and a least element.

According to the relationships between the concept granular computing system and concept lattice, variable threshold concept lattice and fuzzy concept lattice, we can get the following results from Theorem 8.

(1) Let $(U, A, I)$ be a formal context. Then

$$B(P(U), P(A), *, *) = \{(X, B), X^* = B, B^* = X\}$$

is a complete lattice.

(2) Let $\boldsymbol{L} = (L, \vee, \wedge, \otimes, \to, 0, 1)$ be a complete residuated lattice, and $(U, A, \tilde{I})$ be a $L$-fuzzy formal context. Then

$$B(L^U, L^A, +, +) = \{(\tilde{X}, \tilde{B}) : \tilde{X}^+ = \tilde{B}, \tilde{B}^+ = \tilde{X}\} = L_f(U, A, \tilde{I})$$

is a lattice with the great and the least elements.

(3) Let $\boldsymbol{L} = (L, \vee, \wedge, \otimes, \to, 0, 1)$ be a complete residuated lattice, and $(U, A, \tilde{I})$ be a fuzzy formal context. Then

$$B(P(U), P(A), \#, \#) = \{(X, B) : X^\# = B, B^\# = X\}$$

is a lattice with the great and the least elements, and any element in it is called a crisp-crisp variable threshold concept, for simply, variable threshold concept.

(4) Let $\boldsymbol{L}=(L,\vee,\wedge,\otimes,\rightarrow,0,1)$ be a complete residuated lattice, and $(U,A,\tilde{I})$ be a fuzzy formal context. Then

$$B(P(U),L^A,\Delta,\nabla)=\{(X,\tilde{B}):X^{\Delta}=\tilde{B},\tilde{B}^{\nabla}=X\}$$

is a lattice with the great and the least elements, and any element in it is called a crisp-fuzzy variable threshold concept, for simply, variable threshold concept.

(5) Let $\boldsymbol{L}=(L,\vee,\wedge,\otimes,\rightarrow,0,1)$ be a complete residuated lattice, and $(U,A,\tilde{I})$ be a fuzzy formal context. Then

$$B(L^U,P(A),\nabla,\Delta)=\{(\tilde{X},B):\tilde{X}^{\nabla}=B,B^{\Delta}=\tilde{X}\}$$

is a lattice with the great and the least elements, and any element in it is called a fuzzy-crisp variable threshold concept, for simply, variable threshold concept.

**Example 1.** Table 1 shows a fuzzy formal context with $U=\{1,2,3,4\}$ being a set of objects and $A=\{a,b,c,d\}$ being a set of attributes.

**Table 1.** The fuzzy formal context $(U,A,\tilde{I})$

| U | a | B | C | d |
|---|---|---|---|---|
| 1 | 0.5 | 1.0 | 0.7 | 0.5 |
| 2 | 0.6 | 0.7 | 1.0 | 0.5 |
| 3 | 1.0 | 0.9 | 1.0 | 0.1 |
| 4 | 1.0 | 0.9 | 0.9 | 0.1 |

We take Luksiewicz implication operators [15,33,34]

$$a\rightarrow_L b=\begin{cases}1, & a\le b,\\ 1-a+b, & a>b.\end{cases}$$

Then the corresponding adjoin operator is:

$$a\otimes_L b=(a+b-1)\vee 0\,.$$

It is easy to prove that $\boldsymbol{L}=([0,1],\vee,\wedge,\otimes_L,\rightarrow_L,0,1)$ is a residuated complete lattice.

For the fuzzy formal context $(U,A,\tilde{I})$ given in Table 1, take $\delta=1$. Then for any $X\in P(U)$ and $B\in P(A)$, by formula (1) and (2) we can get $X^{\#}$ and $B^{\#}$. Thus, any crisp-crisp variable threshold concept $(X,B)$ satisfying $X^{\#}=B$ and $B^{\#}=X$ can be obtained. Table 2 shows all crisp-crisp variable threshold concepts.

Analogously, for $\delta=1$, $X\in P(U)$ and $\tilde{B}\in F(A)$, we can get all crisp-fuzzy variable threshold concepts by formula (7) and (8). Table 3 shows all crisp-fuzzy variable threshold concepts. And for $\delta = 1$, $\tilde{X} \in F(U)$ and $B \in P(A)$, by formula (9) and (10), we can obtain all fuzzy-crisp variable threshold concepts, see Table 4.

Fig. 1 depicts the three kinds of corresponding variable threshold concept lattices. For simplicity, a set is denoted by listing its elements in sequence. For example, the set $\{1,2,3,4\}$ is denoted by 1234.

**Table 2.** The crisp-crisp variable threshold concepts for $\delta = 1$

| X | B |
|---|---|
| ∅ | {abcd} |
| {3} | {ac} |
| {34} | {a} |
| {23} | {c} |

**Table 3.** The crisp-fuzzy variable threshold concepts for $\delta = 1$

| X | a | b | c | d |
|---|---|---|---|---|
| ∅ | 1.0 | 1.0 | 1.0 | 1.0 |
| {3} | 1.0 | 0.9 | 1.0 | 0.1 |
| {34} | 1.0 | 0.9 | 0.9 | 0.1 |
| {2} | 0.6 | 0.7 | 1.0 | 0.5 |
| {23} | 0.6 | 0.7 | 1.0 | 0.1 |
| {234} | 0.6 | 0.7 | 0.9 | 0.1 |
| {1} | 0.5 | 1.0 | 0.7 | 0.5 |
| {134} | 0.5 | 0.9 | 0.7 | 0.1 |
| {12} | 0.5 | 0.7 | 0.7 | 0.5 |
| {1234} | 0.5 | 0.7 | 0.7 | 0.1 |

**Table 4.** The fuzzy-crisp variable threshold concepts for $\delta = 1$

| B | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| ∅ | 1.0 | 1.0 | 1.0 | 1.0 |
| {c} | 0.7 | 1.0 | 1.0 | 0.9 |
| {b} | 1.0 | 0.7 | 0.9 | 0.9 |
| {bc} | 0.7 | 0.7 | 0.9 | 0.9 |
| {a} | 0.5 | 0.6 | 1.0 | 1.0 |
| {ac} | 0.5 | 0.6 | 1.0 | 0.9 |
| {abc} | 0.5 | 0.6 | 0.9 | 0.9 |
| {abcd} | 0.5 | 0.6 | 0.1 | 0.1 |

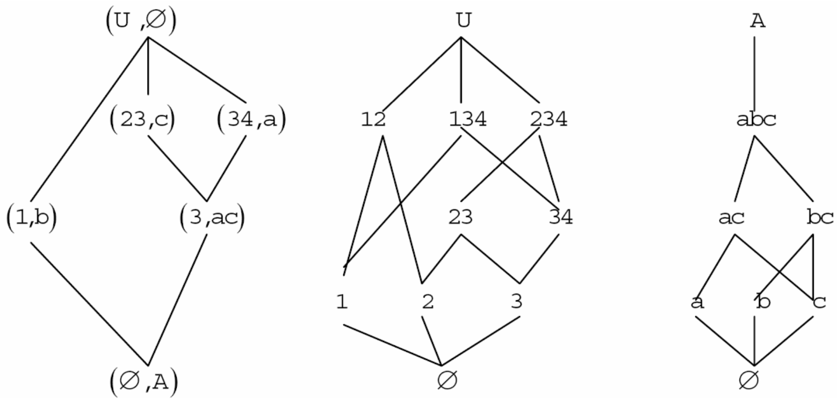


**Fig. 1.** The corresponding variable threshold concept lattices shown in Table 2-Table 4

## 5 Sufficiency and Necessity Information Granules

In order to reflect the granular idea of the concept granular computing system, we introduce information granules.

**Definition 8.** Let $(L_1, L_2, G, H)$ be a concept granular computing system. Note that

$$\mathbb{G}_1 = \{(a,b) : b \le G(a), a \le H(b)\},$$

$$\mathbb{G}_2 = \{(a,b) : G(a) \le b, H(b) \le a\}.$$

If $(a,b) \in \mathbb{G}_1$, we call $(a,b)$ a necessity information granule of the concept granular computing system, and call $b$ the necessity attribute of $a$. Then $\mathbb{G}_1$ is the set of all necessity information granules of the concept granular computing system. (See Fig. 2.).

If $(a,b) \in \mathbb{G}_2$, we call $(a,b)$ a sufficiency information granule of the concept granular computing system, and call $b$ the sufficiency attribute of $a$. Then $\mathbb{G}_2$ is the set of all sufficiency information granules of the concept granular computing system. (See Fig. 3).

If $(a,b) \in \mathbb{G}_1 \cup \mathbb{G}_2$, we call $(a,b)$ an information granule of the concept granular computing system. Then $\mathbb{G}_1 \cup \mathbb{G}_2$ is the set of all information granules of the concept granular computing system.

If $(a,b) \in \mathbb{G}_1 \cap \mathbb{G}_2$, then the pair $(a,b)$ satisfies $b = G(a), a = H(b)$, we call $(a,b)$ a sufficiency and necessity information granule of the concept granular computing system, and call $b$ the sufficiency and necessity attribute of $a$ .Then a sufficiency and necessity information granule is actually a concept of a concept granular computing system defined in Definition 7.

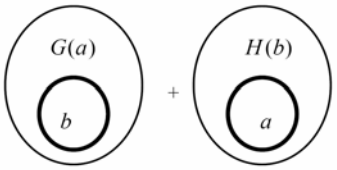


**Fig. 2.** Necessity information granule $(a,b)$

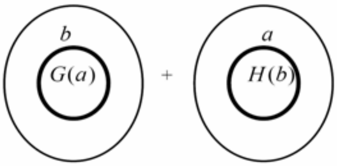


**Fig. 3.** Sufficiency information granule $(a,b)$

If $(a,b) \notin \mathbb{G}_1 \cap \mathbb{G}_2$, we call $(a,b)$ a contradiction information granule.

**Theorem 9.** Let $\mathbb{G}_1$ be a necessity information granule set. For any $(a_1,b_1),(a_2,b_2) \in \mathbb{G}_1$, we define the infimum and the supremum operators on $\mathbb{G}_1$ as follows:

$$(a_1,b_1) \wedge (a_2,b_2) = (a_1 \wedge a_2, G \circ H(b_1 \vee b_2)),$$

$$(a_1,b_1) \vee (a_2,b_2) = (H \circ G(a_1 \vee a_2), b_1 \wedge b_2).$$

Then $\mathbb{G}_1$ is closed under the infimum and supremum operators.

*Proof.* Suppose $(a_1,b_1),(a_2,b_2) \in \mathbb{G}_1$. Then

$$b_1 \le G(a_1), b_2 \le G(a_2), \text{ and } a_1 \le H(b_1), a_2 \le H(b_2).$$

Thus,

$$a_1 \wedge a_2 \le H(b_1) \wedge H(b_2) = H(b_1 \vee b_2) = H \circ G \circ H(b_1 \vee b_2),$$

$$G \circ H(b_1 \vee b_2) = G(H(b_1) \wedge H(b_2)) \le G(a_1 \wedge a_2).$$

Therefore, $(a_1,b_1) \wedge (a_2,b_2)$ is a necessity information granule. Similarly, we can prove that $(a_1,b_1) \vee (a_2,b_2)$ is a necessity information granule.

**Theorem 10.** Let $\mathbb{G}_2$ be a sufficiency information granule set. For any $(a_1,b_1),(a_2,b_2) \in \mathbb{G}_2$, we define the infimum and the supremum operators on $\mathbb{G}_2$ as follows:

$$(a_1,b_1) \wedge (a_2,b_2) = (a_1 \wedge a_2, G \circ H(b_1 \vee b_2)),$$

$$(a_1,b_1) \vee (a_2,b_2) = (H \circ G(a_1 \vee a_2), b_1 \wedge b_2).$$

Then $\mathbb{G}_2$ is closed under the infimum and supremum operators.

*Proof.* Suppose $(a_1,b_1),(a_2,b_2)\in\mathbb{G}_2$. Then

$$G(a_1)\le b_1, G(a_2)\le b_2, \text{ and } H(b_1)\le a_1, H(b_2)\le a_2 .$$

Thus,

$$H\circ G\circ H(b_1\vee b_2)=H(b_1\vee b_2)=H(b_1)\wedge H(b_2)\le a_1\wedge a_2 ,$$

$$G(a_1\wedge a_2)\le G(H(b_1)\wedge H(b_2))=G\circ H(b_1\vee b_2) .$$

Therefore, $(a_1,b_1)\wedge(a_2,b_2)$ is a sufficiency information granule. Similarly, we can prove that $(a_1,b_1)\vee(a_2,b_2)$ is a sufficiency information granule.

**Example 2.** Given a formal context $(U,A,I)$ as Table 5, where $U=\{1,2,3,4\}$ is the set of objects, and $A=\{a,b,c,d\}$ is a set of attributes.

**Table 5.** The formal context $(U,A,I)$

| $U$ | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 |
| 2 | 1 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 |
| 4 | 1 | 1 | 0 | 0 |

From Table 5, we can get the partial necessity information granules (See Fig. 4) and the partial sufficiency information granules (See Fig. 5).

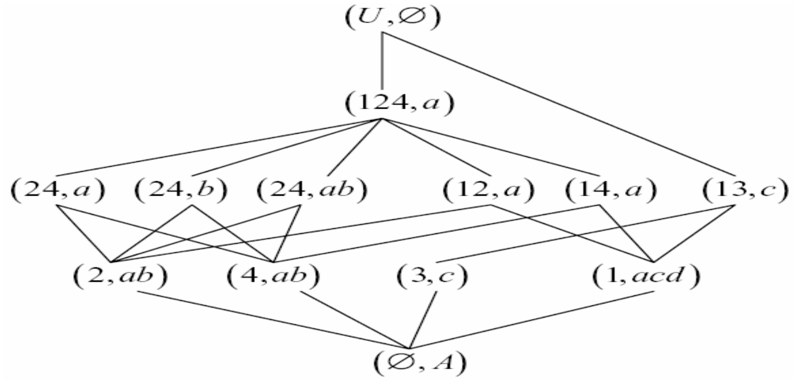


**Fig. 4.** Partial necessity information granule

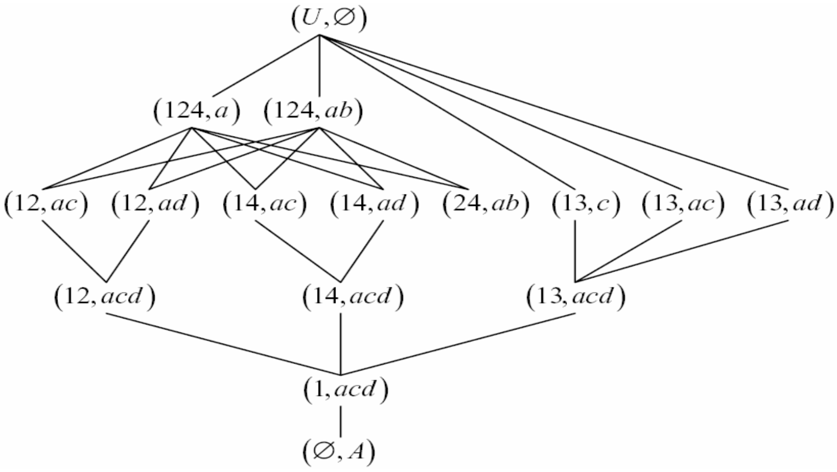


**Fig. 5.** Partial sufficiency information granules

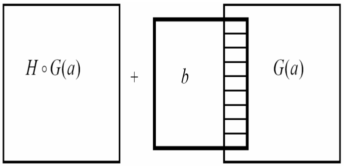


**Fig. 6.** $(H\circ G(a), b\wedge G(a))$

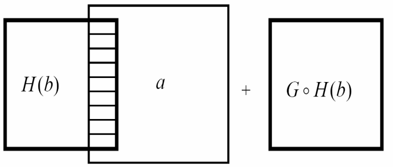


**Fig. 7.** $(a\wedge H(b), G\circ H(b))$

Now we introduce approaches to construct the sufficiency or necessity information granules.

**Theorem 11.** Let $(L_1, L_2, G, H)$ be a concept granular computing system, $\mathbb{G}_1$ is the set of necessity information granules. Then for any $a\in L_1$ and $b\in L_2$, we have

$$(H\circ G(a), b\wedge G(a))\in \mathbb{G}_1 \text{ and } (a\wedge H(b), G\circ H(b))\in \mathbb{G}_1$$

(See Fig. 6 and Fig. 7).

*Proof.* Since $(L_1, L_2, G, H)$ is a concept granular computing system, by Theorem 7 and Definition 8 we have $G\circ H\circ G(a)=G(a)\geq G(a)\wedge b$ and $H(b\wedge G(a))\geq H\circ G(a)\vee H(b)\geq H\circ G(a)$. Thus, $(H\circ G(a), b\wedge G(a))\in G_1$.

Similarly, we can prove $(a\wedge H(b), G\circ H(b))\in G_1$.

**Theorem 12.** Let $(L_1, L_2, G, H)$ be a concept granular computing system, $\mathbb{G}_2$ is the set of sufficiency information granules. Then for any $a\in L_1$ and $b\in L_2$, we have

$$(H\circ G(a), b\vee G(a))\in \mathbb{G}_2 \text{ and } (a\vee H(b), G\circ H(b))\in \mathbb{G}_2$$

(See Fig. 8 and Fig. 9).

*Proof.* Because $(L_1, L_2, G, H)$ is a concept granular computing system, by Theorem 7 and Definition 8 we have $G\circ H\circ G(a)=G(a)\le G(a)\vee b$ and $H(G(a)\vee b)=$ $H\circ G(a)\wedge H(b)\le H\circ G(a)$. Thus, $(H\circ G(a),b\vee G(a))\in G_2$ .Similarly, we can prove that $(a\vee H(b),G\circ H(b))\in G_2$.

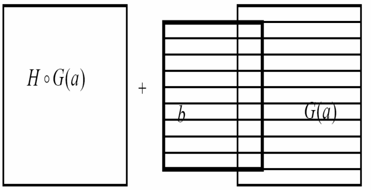


**Fig. 8.** $(H\circ G(a), b\vee G(a))$

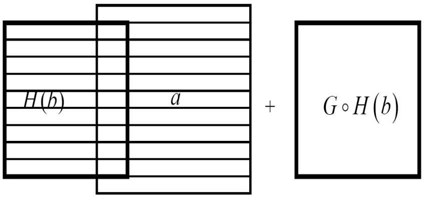


**Fig. 9.** $(a\vee H(b), G\circ H(b))$

**Example 3.** The formal context $(U, A, I)$ is the one given as Example 2.

Then $(P(U), P(A),*,*)$ is a concept granular computing system, and we can obtain all formal concepts which form a concept lattice as Fig. 10.

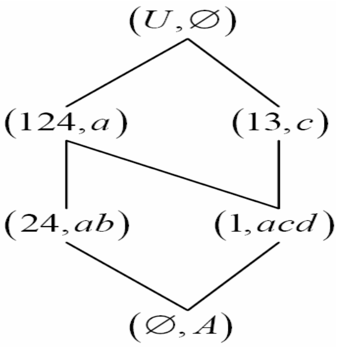


**Fig. 10.** Concept lattice of Example 2

Take $a_0=\{1,4\}$ and $b_0=\{a,b\}$ . Then $(a_0,b_0)$ is a contradiction granule. By Theorem 11, we can calculate that $(H\circ G(a_0), b_0\wedge G(a_0))=(\{1,4\}^{**},\{a,b\}$ $\bigcap\{1,4\}^{*})=(\{1,2,4\},\{a\})$ and $(a_0\wedge H(b_0), G\circ H(b_0))=(\{1,4\}\cap\{a,b\}^{*},$ $\{a,b\}^{**})=(\{4\},\{a,b\})$ are two necessity information granules. Similarly, we can construct the contradiction granule to a sufficiency information granule $(124, ab)$ by using Theorem 12.

From Example 3 we know, for any set of objects and any set of attributes, we can construct necessity or sufficiency information granules by using Theorem 11-Theorem 12, which support a way to construct a sufficiency and necessity information granules, i.e. concepts.

## 6 Iterative Algorithms and Their Optimizations in Concept Granular Computing System

In this section, we establish iterative algorithms to produce concepts from any extent element and intent element.

**Theorem 13.** Let $(L_1, L_2, G, H)$ be a concept granular computing system, and $|L_1| < \infty$ ($|L_1|$ denotes the number of elements in $L_1$). For any $a_1 \in L_1$ and $b_1 \in L_2$, an iterative algorithm is given as follows:

$$\begin{cases} a_n = a_{n-1} \vee H(b_{n-1}), (n \geq 2) \\ b_n = G(a_n) \end{cases} \tag{11}$$

Then for the series of pairs $\{(a_n, b_n)\}_{n\geq 1}$, there exists $n_0 \geq 1$ such that

(1) $(a_{n_0}, b_{n_0}) \in B(L_1, L_2, G, H)$;

(2) For any $(a', b') \in B(L_1, L_2, G, H)$, if $a' \leq a_1 \vee H(b_1)$, then $(a', b') \leq (a_{n_0}, b_{n_0})$.

*Proof.* (1) By the iterative algorithm given by formula (11) we know the sequence $\{a_n\}_{n\geq 1}$ is monotone non-decreasing. Due to $|L_1| < \infty$, there exists a natural number $n_0 \geq 1$ such that for any $n \geq n_0$, we have $a_n = a_{n_0}$. Again using formula (11) we have $a_{n_0} = a_{n_0+1} = a_{n_0} \vee H(b_{n_0})$ and $b_{n_0} = G(a_{n_0})$. Then $a_{n_0} \geq H(b_{n_0})$. By $b_{n_0} = G(a_{n_0})$ we can get that $H(b_{n_0}) = H \circ G(a_{n_0}) \geq a_{n_0}$. Thus $a_{n_0} = H(b_{n_0})$ and $b_{n_0} = G(a_{n_0})$. Therefore, $(a_{n_0}, b_{n_0}) \in B(L_1, L_2, G, H)$.

(2) Suppose $(a', b') \in B(L_1, L_2, G, H)$, and $a' \leq a_1 \vee H(b_1)$. If $(a_1, b_1) \notin B(L_1, L_2, G, H)$, we have $a' \leq a_1 \vee H(b_1) = a_2$. Suppose $a' \leq a_n (n \geq 2)$.

Then $b_n = G(a_n) \leq G(a') = b'$. So $a' = H(b') \leq H(b_n)$. Thus, $a' \leq a_n \vee H(b_n) = a_{n+1}$. By the Inductive law we can obtain that for any $n \geq 2$, $a' \leq a_n$. If $(a_1, b_1) \in B(L_1, L_2, G, H)$, then $a' \leq a_1 \vee H(b_1) = a_1$. Therefore, if $a' \leq a_1 \vee H(b_1)$, we have $a' \leq a_n$ for any $n \geq 1$. For the series of pairs

$(a_n, b_n) \in B(L_1, L_2, G, H)$, by (1) there exists a natural number $n_0 \geq 1$ such that $(a_{n_0}, b_{n_0}) \in B(L_1, L_2, G, H)$. Then by $a' \leq a_n$ for any $n \geq 1$ we have $(a', b') \leq (a_{n_0}, b_{n_0})$.

**Theorem 14.** Let $(L_1, L_2, G, H)$ be a concept granular computing system, and $|L_2| < \infty$. For any $a_1 \in L_1$ and $b_1 \in L_2$, an iterative algorithm is given as follows:

$$\begin{cases} b_n = b_{n-1} \vee G(a_{n-1}), (n \geq 2) \\ a_n = H(b_n) \end{cases} \tag{12}$$

Then for the series of pairs $\{(a_n, b_n)\}_{n\geq 1}$, there exists $n_0 \geq 1$ such that

(1) $(a_{n_0}, b_{n_0}) \in B(L_1, L_2, G, H)$;

(2) For any $(a', b') \in B(L_1, L_2, G, H)$, if $b' \leq b_1 \vee G(a_1)$, then $(a_{n_0}, b_{n_0}) \leq (a', b')$.

*Proof.* (1) By the iterative algorithm given by formula (2.12) we know the sequence $\{b_n\}_{n\geq 1}$ is monotone non-decreasing. Since $|L_2| < \infty$, there exists a natural number $n_0 \geq 1$ such that for any $n \geq n_0$, we have $b_n = b_{n_0}$. Then $b_{n_0} = b_{n_0+1} = b_{n_0} \vee G(a_{n_0})$ and $a_{n_0} = H(b_{n_0})$. Thus, $b_{n_0} \geq G(a_{n_0})$. By $a_{n_0} = H(b_{n_0})$ we can get that $G(a_{n_0}) = G \circ H(b_{n_0}) \geq b_{n_0}$.

So, $b_{n_0} = G(a_{n_0})$ and $a_{n_0} = H(b_{n_0})$. Therefore, $(a_{n_0}, b_{n_0}) \in \mathfrak{B}(L_1, L_2, G, H)$.

(2) Suppose $(a', b') \in B(L_1, L_2, G, H)$, and $b' \leq b_1 \vee G(a_1)$. If $(a_1, b_1) \notin B(L_1, L_2, G, H)$, we have $b' \leq b_1 \vee G(a_1) = b_2$. Suppose $b' \leq b_n (n \geq 2)$. Then $a_n = H(b_n) \leq H(b') = a'$. So $b' = G(a') \leq G(a_n)$. Thus, $b' \leq b_n \vee G(a_n) = b_{n+1}$. By the Inductive law we can obtain that for any $n \geq 2$, $b' \leq b_n$. If $(a_1, b_1) \in B(L_1, L_2, G, H)$, then $b' \leq b_1 \vee G(a_1) = b_1$. Therefore, if $b' \leq b_1 \vee G(a_1)$, we have $b' \leq b_n$ for any $n \geq 1$. For the series of pairs $(a_{n_0}, b_{n_0}) \in B(L_1, L_2, G, H)$, by (1) there exists a natural number $n_0 \geq 1$ such that $(a_{n_0}, b_{n_0}) \in B(L_1, L_2, G, H)$. Then by $b' \leq b_n$ for any $n \geq 1$ we have $(a_{n_0}, b_{n_0}) \leq (a', b')$.

In what follows, we show the iterative algorithms for a formal context and a fuzzy formal context.

**Theorem 15.** Let $(U, A, I)$ be a formal context. For any $X \subseteq U$ and $B \subseteq A$, an iterative algorithm is given as follows:

$$\begin{cases} X_n = X_{n-1} \cup B_{n-1}{}^*, (n \geq 2) \\ B_n = X_n{}^* \end{cases}$$

Then for the series of pairs $\{(X_n, B_n)\}_{n\geq 1}$, there exists $n_0 \geq 1$ such that

(1) $(X_{n_0}, B_{n_0}) \in B(P(U), P(A), *, *)$;

(2) For any $(X', B') \in B(P(U), P(A), *, *)$, if $X' \leq X_1 \cup B_1^*$, then $(X', B') \leq (X_{n_0}, B_{n_0})$.

*Proof.* It is proved by Theorem 1 and Theorem 13.

**Theorem 16.** Let $(U, A, I)$ be a formal context. For any $X \subseteq U$ and $B \subseteq A$, an iterative algorithm is given as follows:

$$\begin{cases} B_n = B_{n-1} \cup X_{n-1}{}^*, (n \geq 2) \\ X_n = B_n{}^* \end{cases}$$

Then for the series of pairs $\{(X_n, B_n)\}_{n\geq 1}$, there exists $n_0 \geq 1$ such that

(1) $(X_{n_0}, B_{n_0}) \in B(P(U), P(A), *, *)$;

(2) For any $(X', B') \in B(P(U), P(A), *, *)$, if $B' \leq B_1 \cup X_1^*$, then $(X_{n_0}, B_{n_0}) \leq (X', B')$.

*Proof.* It is proved by Theorem 1 and Theorem 14.

**Theorem 17.** Let $(U, A, \tilde{I})$ be a $L$-fuzzy formal context. For any $\tilde{X} \subseteq L^U$ and $\tilde{B} \subseteq L^A$,

(1) if an iterative algorithm is given as follows:

$$\begin{cases} \tilde{X}_n = \tilde{X}_{n-1} \cup \tilde{B}_{n-1}{}^+, (n \geq 2) \\ \tilde{B}_n = \tilde{X}_n{}^+ \end{cases}$$

Then for the series of pairs $\{(\tilde{X}_n, \tilde{B}_n)\}_{n\geq 1}$, there exists $n_0 \geq 1$ such that $(\tilde{\tilde{X}}_{n_0}, \tilde{\tilde{B}}_{n_0})$ $\in B(L^U, L^A, +, +)$. And for any $(\tilde{\tilde{X}}', \tilde{\tilde{B}}') \in B(L^U, L^A, +, +)$, if $\tilde{X}' \leq \tilde{X}_1 \cup \tilde{B}_1^+$, then $(\tilde{X}', \tilde{B}') \leq (\tilde{X}_{n_0}, \tilde{B}_{n_0})$;

(2) if an iterative algorithm is given as follows:

$$\begin{cases} \tilde{B}_n = \tilde{B}_{n-1} \cup \tilde{X}_{n-1}{}^+, (n \geq 2) \\ \tilde{X}_n = \tilde{B}_n{}^+ \end{cases}$$

Then for the series of pairs $\{(\tilde{X}_n,\tilde{B}_n)\}_{n\geq 1}$, there exists $n_0 \geq 1$ such that $(\tilde{X}_{n_0},\tilde{B}_{n_0})\in B(L^U,L^A,+,+)$. And for any $(\tilde{X}',\tilde{B}')\in B(L^U,L^A,+,+)$, if $\tilde{B}'\leq \tilde{B}_1 \cup \tilde{X}_1^{+}$, then $(\tilde{X}_{n_0},\tilde{B}_{n_0}) \leq (\tilde{X}',\tilde{B}')$.

*Proof.* It is proved by Theorem 2, Theorem 13 and Theorem 14.

**Example 4.** Let $(U,A,I)$ be a formal context given in Example 2. Take $X_0=\{1,4\}$ and $B_0=\{a,b\}$. $(X_0,B_0)$ is not a formal context. By Theorem 15 and Theorem 16 we can get that $(24,ab)$ and $(124,a)$ are concepts.

## 7 Rough Set Approximations in Formal Concept Analysis

A structure $\boldsymbol{L}=(L,\vee,\wedge,\otimes,\rightarrow,0,1)$ is referred to as a complete involutive residuated lattice if $L$ is a complete residuated lattice and the operator $c: L\rightarrow L$ satisfies $a_1\leq a_2 \Rightarrow a_2^c \leq a_1^c$ and $a^{cc}=a$ for any $a_1,a_2,a\in L$, where $c$ represents the complement operator of any element of $L$.

A $L$-fuzzy formal context $(U,A,\tilde{I})$ is called an involutive $L$-fuzzy formal context if $\boldsymbol{L}=(L,\vee,\wedge,\otimes,\rightarrow,0,1)$ is a complete involutive residuated lattice. Then for any $\tilde{X}\in L^U$ and $\tilde{B}\in L^A$, we define the following operators[ 27,39]:

$$\tilde{X}^{+}(a)=\underset{x\in U}{\wedge}(\tilde{X}(x)\rightarrow \tilde{I}(x,a))$$

$$\tilde{B}^{+}(x)=\underset{a\in A}{\wedge}(\tilde{B}(a)\rightarrow \tilde{I}(x,a))$$

$$\tilde{X}^{\uparrow}(a)=\underset{x\in U}{\vee}(\tilde{X}^{c}(x)\otimes \tilde{I}^{c}(x,a))$$

$$\tilde{B}^{\downarrow}(x)=\underset{a\in A}{\vee}(\tilde{B}^{c}(a)\otimes \tilde{I}^{c}(x,a))$$

$$\tilde{X}^{\Diamond}(a)=\underset{x\in U}{\vee}(\tilde{X}(x)\otimes \tilde{I}(x,a))$$

$$\tilde{X}^{\Box}(a)=\underset{x\in U}{\wedge}(\tilde{I}(x,a)\rightarrow \tilde{X}(x))$$

$$\tilde{B}^{\Diamond}(x)=\underset{a\in A}{\vee}(\tilde{B}(a)\otimes \tilde{I}(x,a))$$

$$\tilde{B}^{\Box}(x)=\underset{a\in A}{\wedge}(\tilde{I}(x,a)\rightarrow \tilde{B}(a))$$

**Definition 10.** Let $(U,A,\tilde{I})$ be an involutive $L$-fuzzy formal context. For any $\tilde{X}\in L^U$, we define

$$\underline{Apri}(\tilde{X})=\tilde{X}^{\uparrow\downarrow} \text{ and } \overline{Apri}(\tilde{X})=\tilde{X}^{++} \tag{13}$$

$\underline{Apri}(\tilde{X})$ and $\overline{Apri}(\tilde{X})$ are referred to as the lower and upper approximations of $\tilde{X}$, respectively. And the operators $\uparrow\downarrow: L^U\rightarrow L^U$ and $++: L^U\rightarrow L^U$ are referred to as the $i$-model lower and upper approximation operators, respectively. The pair $(\underline{Apri}(\tilde{X}),\overline{Apri}(\tilde{X}))$ is referred to as a generalized $i$-model rough fuzzy set.

**Theorem 18.** Let $(U,A,\tilde{I})$ be an involutive $L$-fuzzy formal context. Then for any $\tilde{X},\tilde{Y}\in L^U$, the $i$-model lower and upper approximation operators satisfy the following properties:

($FL_1$) $\underline{Apri}(\tilde{X})=(\overline{Apri}(\tilde{X}^c))^c$;

($FU_1$) $\overline{Apri}(\tilde{X})=(\underline{Apri}(\tilde{X}^c))^c$;

($FL_2$) $\underline{Apri}(\varnothing)=\overline{Apri}(\varnothing)=\varnothing$;

($FU_2$) $\overline{Apri}(U)=\underline{Apri}(U)=U$;

($FL_3$) $\underline{Apri}(\tilde{X}\cap\tilde{Y})\subseteq\underline{Apri}(\tilde{X})\cap\underline{Apri}(\tilde{Y})$;

($FU_3$) $\overline{Apri}(\tilde{X}\cup\tilde{Y})\supseteq\overline{Apri}(\tilde{X})\cup\overline{Apri}(\tilde{Y})$;

($FL_4$) $\tilde{X}\subseteq\tilde{Y}\Rightarrow\underline{Apri}(\tilde{X})\subseteq\underline{Apri}(\tilde{Y})$;

($FU_4$) $\tilde{X}\subseteq\tilde{Y}\Rightarrow\overline{Apri}(\tilde{X})\subseteq\overline{Apri}(\tilde{Y})$;

($FL_5$) $\underline{Apri}(\tilde{X}\cup\tilde{Y})\supseteq\underline{Apri}(\tilde{X})\cup\underline{Apri}(\tilde{Y})$;

($FU_5$) $\overline{Apri}(\tilde{X}\cap\tilde{Y})\subseteq\overline{Apri}(\tilde{X})\cap\overline{Apri}(\tilde{Y})$;

($FL_6$) $\underline{Apri}(\tilde{X})\subseteq\tilde{X}$;

($FU_6$) $\tilde{X}\subseteq\overline{Apri}(\tilde{X})$;

($FL_7$) $\underline{Apri}(\underline{Apri}(\tilde{X}))=\underline{Apri}(\tilde{X})$;

($FU_7$) $\overline{Apri}(\overline{Apri}(\tilde{X}))=\overline{Apri}(\tilde{X})$.

*Proof.* ($FL_1$) and ($FU_1$) show that the approximation operators $\underline{Apri}$ and $\overline{Apri}$ are dual to each other. Then we only need to prove ($FL_i$) or ($FU_i$), by the duality we can easily get ($FU_i$) or ($FL_i$) $(i=1,\cdots,7)$.

For any $\tilde{X}\in L^U$ and $x\in U$,

$$\begin{aligned}(\underline{Apri}(\tilde{X}^c))^c(x)&=((\tilde{X}^c)^{\uparrow\downarrow}(x))^c\\&=\vee_{a\in A}(\vee_{y\in U}(\tilde{X}^{cc}(y)\otimes\tilde{I}^c(y,a))^c\otimes\tilde{I}^c(x,a))^c\\&=\vee_{a\in A}(\wedge_{y\in U}(\tilde{X}(y)\to\tilde{I}(y,a))\otimes\tilde{I}^c(x,a))^c\\&=\wedge_{a\in A}(\wedge_{y\in U}(\tilde{X}(y)\to\tilde{I}(y,a))\to\tilde{I}(x,a))\\&=\tilde{X}^{++}(x)\end{aligned}$$

Thus, ($FU_1$) holds. ($FL_2$) immediately follows by Definition 10.

According to $(\tilde X \cup \tilde Y)^+ = \tilde X^+ \cap \tilde Y^+$, $\tilde X^+ \cap \tilde Y^+ \subseteq \tilde X^+$ and $\tilde X^+ \cap \tilde Y^+ \subseteq \tilde Y^+$, we have

$$\tilde X^{++} \subseteq (\tilde X^+ \cap \tilde Y^+)^+ \text{ and } \tilde Y^{++} \subseteq (\tilde X^+ \cap \tilde Y^+)^+ .$$

Thus,

$$\overline{Apri}(\tilde X) \cup \overline{Apri}(\tilde Y) = \tilde X^{++} \cup \tilde Y^{++} \subseteq (\tilde X^+ \cap \tilde Y^+)^+ = \overline{Apri}(\tilde X \cup \tilde Y),$$

from which we can get ($FU_3$).

($FL_4$) follows immediately from $\tilde X_1 \subseteq \tilde X_2 \Rightarrow \tilde X_2^{\uparrow} \subseteq \tilde X_1^{\uparrow}$ and $\tilde B_1 \subseteq \tilde B_2 \Rightarrow \tilde B_2^{\downarrow} \subseteq \tilde B^{\downarrow}$.

Since $\tilde X^{++} \cap \tilde Y^{++} = (\tilde X^+ \cup \tilde Y^+)^+$, $\tilde X^+ \subseteq (\tilde X \cap \tilde Y)^+$ and $\tilde Y^+ \subseteq (\tilde X \cap \tilde Y)^+$, we have

$$\tilde X^{++} \supseteq (\tilde X \cap \tilde Y)^{++} \text{ and } \tilde Y^{++} \supseteq (\tilde X \cap \tilde Y)^{++} .$$

Therefore,

$$\overline{Apri}(\tilde X) \cap \overline{Apri}(\tilde Y) = \tilde X^{++} \cap \tilde Y^{++} \supseteq (\tilde X \cap \tilde Y)^{++} = \overline{Apri}(\tilde X \cap \tilde Y) .$$

Thus, ($FU_5$) holds.

($FL_6$) follows directly by $\tilde X \subseteq \tilde X^{\uparrow\downarrow}$ and $\tilde B \subseteq \tilde B^{\downarrow\uparrow}$.

Since $\underline{Apri}(\underline{Apri}(\tilde X)) = (\tilde X^{\uparrow\downarrow})^{\uparrow\downarrow}$ and $\tilde X^{\uparrow\downarrow\uparrow} = \tilde X^{\uparrow}$, we can get ($FL_7$).

**Definition 11.** Let $(U, A, \tilde I)$ be an involutive $L$-fuzzy formal context. For any $\tilde X \in L^U$, we define the lower and upper approximations of $\tilde X$ as follows:

$$\underline{Aprii}(\tilde X) = \tilde X^{\Box\Diamond} \text{ and } \overline{Apri}(\tilde X) = \tilde X^{\Diamond\Box} \tag{14}$$

Then the operators $\Box\Diamond : L^U \to L^U$ and $\Diamond\Box : L^U \to L^U$ are referred to as the *ii*-model lower and upper approximation operators, respectively. The pair $(\underline{Aprii}(\tilde X)$, $\underline{Aprii}(\tilde X))$ is referred to as a generalized *ii*-model rough fuzzy set.

**Theorem 19.** Let $(U, A, \tilde I)$ be an involutive $L$-fuzzy formal context. Then for any $\tilde X, \tilde Y \in L^U$, the *ii*-model lower and upper approximation operators satisfy the following properties:

($FL'_1$) $\underline{Aprii}(\tilde X) = (\overline{Aprii}(\tilde X^c))^c$;

($FU'_1$) $\overline{Aprii}(\tilde X) = (\underline{Aprii}(\tilde X^c))^c$;

($FL'_2$) $\underline{Aprii}(\emptyset) = \overline{Aprii}(\emptyset) = \emptyset$;

($FU'_2$) $\overline{Aprii}(U) = \underline{Aprii}(U) = U$;

$(FL'_3)$ $\underline{Aprii}(\tilde{X} \cap \tilde{Y}) \subseteq \underline{Aprii}(\tilde{X}) \cap \underline{Aprii}(\tilde{Y})$ ;

$(FU'_3)$ $\overline{Aprii}(\tilde{X} \cup \tilde{Y}) \supseteq \overline{Aprii}(\tilde{X}) \cup \overline{Aprii}(\tilde{Y})$ ;

$(FL'_4)$ $\tilde{X} \subseteq \tilde{Y} \Rightarrow \underline{Aprii}(\tilde{X}) \subseteq \underline{Aprii}(\tilde{Y})$ ;

$(FU'_4)$ $\tilde{X} \subseteq \tilde{Y} \Rightarrow \overline{Aprii}(\tilde{X}) \subseteq \overline{Aprii}(\tilde{Y})$ ;

$(FL'_5)$ $\underline{Aprii}(\tilde{X} \cup \tilde{Y}) \supseteq \underline{Aprii}(\tilde{X}) \cup \underline{Aprii}(\tilde{Y})$ ;

$(FU'_5)$ $\overline{Aprii}(\tilde{X} \cap \tilde{Y}) \subseteq \overline{Aprii}(\tilde{X}) \cap \overline{Aprii}(\tilde{Y})$ ;

$(FL'_6)$ $\underline{Aprii}(\tilde{X}) \subseteq \tilde{X}$ ;

$(FU'_6)$ $\tilde{X} \subseteq \overline{Aprii}(\tilde{X})$ ;

$(FL'_7)$ $\underline{Aprii}(\underline{Aprii}(\tilde{X})) = \underline{Aprii}(\tilde{X})$ ;

$(FU'_7)$ $\overline{Aprii}(\overline{Aprii}(\tilde{X})) = \overline{Aprii}(\tilde{X})$ .

*Proof*. We still prove $(FL'_i)$ or $(FU'_i)$, by the duality we can easily get $(FU'_i)$ or $(FL'_i)$ $(i = 1, \cdots, 7)$ .

For any $\tilde{X} \in L^U$ ,

$$(\overline{Aprii}(\tilde{X}^c))^c = ((\tilde{X}^c)^{\Diamond\Box})^c = (((\tilde{X}^c)^{c\Box c})^{\Box})^c$$
$$= (\tilde{X}^{\Box c\Box})^c = \tilde{X}^{\Box c\Box c} = (\tilde{X}^{\Box})^{c\Box c} = \tilde{X}^{\Box\Diamond} = \underline{Aprii}(\tilde{X})$$

Thus, $(FU'_1)$ holds.

For any $x \in U$ ,

$$\begin{aligned}\overline{Aprii}(\varnothing)(x) &= \varnothing^{\Diamond\Box}(x) \\ &= \wedge_{a \in A}(\tilde{I}(x,a) \to \varnothing^{\Diamond}(a)) \\ &= \wedge_{a \in A}(\tilde{I}(x,a) \to (\vee_{y \in U}(\varnothing(y) \otimes \tilde{I}(y,a)))) \\ &= \wedge_{a \in A}(\tilde{I}(x,a) \to 0) \\ &= 0.\end{aligned}$$

By $\tilde{X}^{\Box\Diamond} \subseteq \tilde{X} \subseteq \tilde{X}^{\Diamond\Box}$ we can get that $\underline{Aprii}(\varnothing) = \varnothing$ .

For any $\tilde{X}, \tilde{Y} \in L^U$ , according to

$$(\tilde{X} \cap \tilde{Y})^{\Box\Diamond} = (\tilde{X}^{\Box} \cap \tilde{Y}^{\Box})^{\Diamond}, \tilde{X}^{\Box} \cap \tilde{Y}^{\Box} \subseteq \tilde{X}^{\Box} \text{ and } \tilde{X}^{\Box} \cap \tilde{Y}^{\Box} \subseteq \tilde{Y}^{\Box},$$

we have

$$(\tilde{X}^{\Box} \cap \tilde{Y}^{\Box})^{\Diamond} \subseteq \tilde{X}^{\Box\Diamond} \text{ and } (\tilde{X}^{\Box} \cap \tilde{Y}^{\Box})^{\Diamond} \subseteq \tilde{Y}^{\Box\Diamond},$$

from which we can get $(FL'_3)$.

($FL'_4$) follows immediately by $\tilde{X}_1 \subseteq \tilde{X}_2 \Rightarrow \tilde{X}_1^{\Diamond} \subseteq \tilde{X}_2^{\Diamond}, \tilde{X}_1^{\square} \subseteq \tilde{X}_2^{\square}$.

By $\tilde{X}^{\square} \subseteq (\tilde{X} \cup \tilde{Y})^{\square}$, $\tilde{Y}^{\square} \subseteq (\tilde{X} \cup \tilde{Y})^{\square}$, $\tilde{X}^{\square\Diamond} \subseteq (\tilde{X} \cup \tilde{Y})^{\square\Diamond}$ and $\tilde{Y}^{\square\Diamond} \subseteq (\tilde{X} \cup \tilde{Y})^{\square\Diamond}$ we can get ($FL'_5$).

($FL'_6$) follows directly by $\tilde{X}^{\square\Diamond} \subseteq \tilde{X} \subseteq \tilde{X}^{\Diamond\square}$.

Since $\underline{Aprii}(\underline{Aprii}(\tilde{X})) = (\tilde{X}^{\square\Diamond})^{\square\Diamond}$ and $\tilde{X}^{\square\Diamond\square} = \tilde{X}^{\square}$, we can get ($FL'_7$).

The approach of rough set approximation in concept analysis gives a way for studying concept lattice via rough set.

## 8 Conclusions

Since FCA was introduced by Wille in 1982, many researches studied it from various points and extended it to more complex situations such as a *L*-fuzzy formal context which is appropriated to the real world. In this paper, a concept granular computing system is established based on the study of concept lattice and *L*-fuzzy concept lattice. Relationships between this system and concept lattice, variable threshold concept lattice and fuzzy concept lattice and properties of the system are discussed. In order to reflect different relations between a set of objects and a set of attributes, sufficiency information granules and necessity information granules are defined. Properties of them are then studied. Later, iterative algorithms for constructing concepts for any extent element or intent element are introduced, and the optimization of the iterative algorithms are investigated. Finally, set approximations in FCA are studied, which shows a way to study FCA by using the theory of rough set.

Learning and application for concepts is the key question in the field of artificial intelligence. In order to process information via computers, a kind of mathematical model needs to be built. This paper is a try to build concept granular computing system by introducing an algebra structure. It has more benefits for further studies, such as concept generalization and specialization, sufficiency and necessity concept, more generalized concept and more special concept. And using this framework, a researcher can conclude some axiomic characterizations from various kinds of concept systems. Therefore, this model may supply an important tool for the further study of the formation and learning of concepts.

**Acknowledgments.** This paper is granted by the National Science of Foundation Funds No. 60703117 and No. 60673096. The authors thank for all people from the research team in Xi'an Jiaotong University and their contributions to this paper.

## References

1. Belohlavek, R.: Fuzzy Galois connections. Mathematical Logic Quarterly 45(4), 497–504 (1999)
2. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic/Plenum Publishers, New York (2002)

3. Belohlavek, R.: Concept lattices and order in fuzzy logic. Annals of Pure and Applied Logic 128, 277–298 (2004)
4. Belohlavek, R., Sklenar, V., Zacpal, J.: Crisply generated fuzzy concepts. In: Ganter, B., Godin, R. (eds.) ICFCA 2005. LNCS, vol. 3403, pp. 269–284. Springer, Heidelberg (2005)
5. Belohlavek, R., Vychodil, V.: Reducing the size of fuzzy concept lattices by hedges. In: The IEEE International Conference on Fuzzy Systems, USA, pp. 663–668 (2005)
6. Belohlavek, R., Vychodil, V.: What is a fuzzy concept lattice? In: Proceedings of CLA 2005, 3rd International Conference on Concept Lattices and their Applications, Olomouc, Czech Republic, pp. 34–45 (2005)
7. Belohlavek, R.: A note on variable threshold concept lattices: threshold-based operators are reducible to classical concept-forming operators. Information Sciences 177, 3186–3191 (2007)
8. Burusco, A., Fuentes-Gonzales, R.: The study of the L-fuzzy concept lattice. Mathware & Soft Computing I(3), 209–218 (1994)
9. Burusco, A., Fuentes-Gonzales, R.: Concept lattices defined from implication operators. Fuzzy Sets and Systems 114, 431–436 (2000)
10. Burusco, A., Fuentes-Gonzales, R.: Construction of the L-fuzzy concept lattice. Fuzzy Sets and Systems 97, 109–114 (1998)
11. Chaudron, L., Maille, N.: Generalized formal concept analysis. In: Ganter, B., Mineau, G.W. (eds.) ICCS 2000. LNCS, vol. 1867, pp. 357–370. Springer, Heidelberg (2000)
12. Chen, D.G., Zhang, W.X., Yeung, D., et al.: Rough approximations on a completely distributive lattice with applications to generalized rough sets. Information Science 176(13), 1829–1948 (2006)
13. Deogun, J.S., Saqer, J.: Monotone concepts for formal concept analysis. Discrete Applied Mathematics 144, 70–78 (2004)
14. Dubois, D., Prade, H.: Twofold fuzzy sets and rough sets—some issues in knowledge representation. Fuzzy Sets and Systems 23, 3–18 (1987)
15. Duntsch, I., Gediga, G.: Modal-style operators in qualitative data analysis. In: Proc. 2002 IEEE Inter. Conf. on Data Mining, pp. 155–162 (2002)
16. Elloumi, S., Jaam, J., Hasnah, A., et al.: A multi-level conceptual data reduction approach based in the Lukasiewicz implication. Information Sciences 163(4), 253–262 (2004)
17. Fan, S.Q., Zhang, W.X., Xu, W.: Fuzzy inference based on fuzzy concept lattice. Fuzzy Sets and Systems 157, 3177–3187 (2006)
18. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999)
19. Georgescu, G., Popescu, A.: Non-dual fuzzy connections. Archive for Mathematic Logic 43(8), 1009–1039 (2004)
20. Hobbs, J.R.: Granularity. In: Proc of IJCAI, Los Angeles, pp. 432–435 (1985)
21. Hu, K., Sui, Y., Lu, Y., et al.: Concept approximation in concept lattice. In: Cheung, D., Williams, G.J., Li, Q. (eds.) PAKDD 2001. LNCS (LNAI), vol. 2035, pp. 167–173. Springer, Heidelberg (2001)
22. Jaoua, A., Elloumi, S.: Galois connection, formal concept and Galois lattice in real binary relation. Journal of Systems and Software 60(2), 149–163 (2002)
23. Kent, R.E.: Rough concept analysis: a synthesis of rough sets and formal concept analysis. Fund. Inform. 27, 169–181 (1996)
24. Krajci, S.: Cluster based efficient generation of fuzzy concepts. Neural Network World 5, 521–530 (2003)

25. Latiri, C.C., Elloumi, S., Chevallet, J.P., et al.: Extension of fuzzy Galois connection for information retrieval using a fuzzy quantifier. In: ACS/IEEE International Conference on Computer Systems and Applications, Tunis, Tunisia (2003)
26. Ma, J.M., Zhang, W.X., Leung, Y., et al.: Granular computing and dual Galois connection. Information Science 177, 5365–5377 (2007)
27. Morsi, N.N., Yakout, M.M.: Axiomatic for fuzzy rough sets. Fuzzy Sets and Systems 100, 327–342 (1998)
28. Popescu, A.: A general approach to fuzzy concepts. Mathematical Logic Quarterly 50(3), 1–17 (2001)
29. Qiu, G.F.: Learning models based on formal context. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS, vol. 4481, pp. 419–426. Springer, Heidelberg (2007)
30. Saquer, J., Deogun, J.S.: Formal rough concept analysis. In: Zhong, N., Skowron, A., Ohsuga, S. (eds.) RSFDGrC 1999. LNCS, vol. 1711, pp. 91–99. Springer, Heidelberg (1999)
31. Shao, M.W., Liu, M., Zhang, W.X.: Set approximations in fuzzy formal concept analysis. Fuzzy Sets and Systems 158, 2627–2640 (2007)
32. Skowron, A., Stepaniuk, J.: Information granules: towards foundations of granular computing. International Journal of Intelligent Systems 16, 57–85 (2001)
33. Wang, G.J.: Non-Classical Mathematical Logic and Approximate Reasoning. Science Press, Beijing (2000)
34. Wang, G.J., Zhang, W.X.: Consistency degree of finite theories in Lukasiwicz propositional fuzzy logic. Fuzzy Sets and Systems 149, 275–284 (2005)
35. Wang, H., Zhang, W.X.: Relationships between concept lattice and rough set. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS, vol. 4029, pp. 538–547. Springer, Heidelberg (2006)
36. Ward, M., Dilworth, R.P.: Residuated lattices. Transactions of the American Mathematical Society 45, 335–354 (1939)
37. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)
38. Wolff, K.E.: A conceptual view of knowledge bases in rough set theory. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS (LNAI), vol. 2005, pp. 220–228. Springer, Heidelberg (2001)
39. Wu, W.Z., Mi, J.S., Zhang, W.X.: Generalized fuzzy rough sets. Information Sciences 151, 263–282 (2003)
40. Yahia, S., Jaoua, A.: Discovering knowledge from fuzzy concept lattice. In: Kandel, A., Last, M., Bunke, H. (eds.) Data Mining and Computational Intelligence, pp. 167–190. Physica-Verlag (2001)
41. Yao, Y.Y.: Information granulation and rough set approximation. International Journal of Intelligent Systems 16, 87–104 (2001)
42. Yao, Y.Y.: Concept lattices in rough set theory. In: Proc. 2004 Annu. Meeting of the North American Fuzzy Information Processing Society, pp. 796–801 (2004)
43. Yao, Y.Y.: A comparative study of formal concept analysis and rough set theory in data analysis. In: Tsumoto, S., Słowiński, R., Komorowski, J., Grzymała-Busse, J.W. (eds.) RSCTC 2004. LNCS, vol. 3066, pp. 59–68. Springer, Heidelberg (2004)
44. Yao, Y.Y., Chen, Y.: Rough set approximations in formal concept analysis. In: 2004 Annu. Meeting of the North American Fuzzy Information Processing Society, pp. 73–78 (2004)
45. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, N., Yager, R. (eds.) Advances in Fuzzy Set Theory and Application, pp. 3–18. Northholland, Amsterdam (1979)

46. Zadeh, L.A.: Fuzzy logic-computing with words. IEEE Transactions on Fuzzy Systems 4, 103–111 (1996)
47. Zadeh, L.A.: Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 19, 111–127 (1997)
48. Zhang, W.X., Leung, Y., Wu, W.Z.: Information Systems and Knowledge Discovery. Science Press, Beijing (2003)
49. Zhang, W.X., Qiu, G.F.: Uncertain Decision Making Based on Rough Sets. Tsinghua University Press, Beijing (2003)
50. Zhang, W.X., Wei, L., Qi, J.J.: Attribute reduction theory and approach to concept lattice. Science in China Series F-Information Sciences, vol. 48(b), pp. 713–726 (2005)

# Interpretability of Fuzzy Information Granules

Corrado Mencar

University of Bari, Italy
mencar@di.uniba.it

**Abstract.** Human-Centric Information Processing requires tight communication processes between users and computers. These two actors, however, traditionally use different paradigms for representing and manipulating information. Users are more inclined in managing perceptual information, usually expressed in natural language, whilst computers are formidable number-crunching systems, capable of manipulating information expressed in precise form. Fuzzy information granules could be used as a common interface for communicating information and knowledge, because of their ability of representing perceptual information in a computer manageable form. Nonetheless, this connection could be established only if information granules are interpretable, i.e. they are semantically co-intensive with human knowledge. Interpretable information granulation opens several methodological issues, regarding the representation and manipulation of information granules, the interpretability constraints and the granulation processes. By taking into account all such issues, effective Information Processing systems could be designed with a strong Human-Centric imprint.

## 1 Introduction

Human-Centered Computing (HCC) is a new field embracing all the methodologies that apply to applications in which people directly interact with computer technologies. Thus, HCC refers to a modern way of tackling computing issues by taking into account user needs and constraints [1, Ch. 1].

We stress the importance of communication between users and machines, the former acting as producers/consumers of information and the latter being involved in the concrete task of information processing. Besides, we observe the different paradigms for interpreting and manipulating information by users and computers. Users are indeed more inclined in managing perceptual information, usually expressed in natural language, whilst computers are formidable number-crunching systems, capable of manipulating information expressed in precise form.

The “semantic gap” between users and machines is apparent. Quite often, this gap is filled by users, which support the effort of translating perceptual information into computer-understandable forms and interpreting computer results. This approach requires technically skilled users and prevents computers to be easily used by other people who may take full advantage from more “humanized” machines (e.g. physicians, managers, decision makers, etc.) [2].

From the last decade, however, a paradigm shift – the so-called Human Centric Computing – is in act [3]. The great enhancement of computing technologies, as well as the birth and consolidation of new computing models (e.g. Granular Computing) are encouraging the development of novel techniques and methods that enable

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 95 – 118.
springerlink.com

computers to fill the semantic gap. In Information Processing, this paradigm shift has a great impact: users could provide input information in a perceptual form (e.g. in natural language) and could read and understand the subsequent results even without specific technical skills. In a nutshell, Human Centric Information Processing (HCIP) accounts users[1] as initiators of information processing as well as final recipients of the subsequent results. Through HCIP, machine intelligence increases dramatically and enables a more pervasive diffusion of computing.

The semantic gap between users and machines is due to the different nature of information that is represented and manipulated by these two actors. Two important features distinguish perceptual information from precise information, as pointed out by Zadeh: granularity and fuzziness (or graduality) [4-6].

Granularity refers to the property of information to refer to a clump of objects instead of a single one. Objects in a granular information (or information granule) are related by some proximity relation (in a wide sense). Representation and manipulation of information granules fall within Granular Computing, a key computing paradigm for HCIP [7,8].

Information granularity is required for economizing the representation of complex situations and phenomena, where precision is not necessary. Thus, granular information is used in mental processing of perceptual information. Furthermore, information granularity enables the use of natural language to describe facts. Most natural language sentences indeed represent granular information (e.g. “there is warm temperature in the room” does not specify any precise degree). This form of information could be sufficient for users to make decisions (e.g. turn-on the air conditioner), since in most cases users are unable to get more precise information (e.g. the exact temperature distribution of the room) nor they are interested.

Fuzziness is strictly related to information granules. According to this property, the membership of an object to an information granule is gradual rather than dichotomic. Fuzziness reflects the fact that natural phenomena are continue rather than discrete, and they are perceived by people with continuity. It is hence natural to assume that mental percepts reflect the graduality of the perceived phenomena. As a consequence, the semantics of natural language terms, which are used to symbolically describe perceptual information, embodies the fuzziness property.

Fuzzy information granules define information with granularity and fuzziness properties. They capture the key features of perceptual information and are naturally represented in natural language. Hence they constitute the basic underpinning for HCIP.

Fuzzy information granules should be also represented in computers and some mathematical machinery should be available in order to process this type of information. Fuzzy Set Theory (FST) provides such a machinery. Fuzzy Information Granules are represented as fuzzy sets, and fuzzy set operators are used to elaborate information.

Through FST, information processing can take form. Users can input perceptual information in natural language or in similar forms to a computer program. Such perceptual information is converted into fuzzy sets (a process called “m-precisiation” [9,10], where ‘m’ stands for ‘machine’). Fuzzy sets are processed according to program objectives, and results are usually represented as fuzzy sets. Finally, resulting

[1] Thorough this Chapter users are intended as human users.

fuzzy sets are converted into linguistic forms, according to a “m-imprecisiation” mechanism. Users are unaware that computers use numeric/symbolic operations, and computers are unaware that what they are elaborating are actually representations of perceptual information. Fuzzy information granules constitute a communication interface between two very different worlds (Fig. 1).

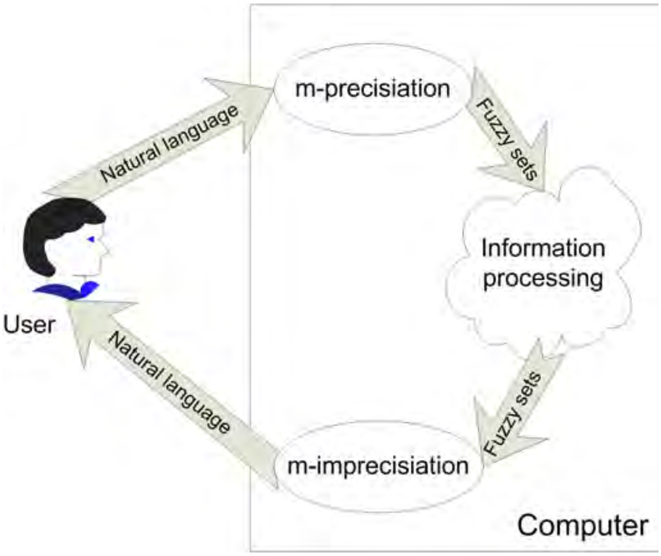


**Fig. 1.** Information flow in HCIP

We should note that without HCIP the same flow of information processing takes place, but with some remarkable differences: precisiation and imprecisiation are carried out by users (they are denoted with “h-precisiation” and “h-imprecisiation” in the Zadeh’s notation, where ‘h’ stands for ‘human’) and information processing is carried out according to classical numerical/symbolic techniques.

Human-Centered information processing is not immune to problems directly deriving from the objects of processing, i.e. fuzzy information granules. More precisely, the processes of m-precisiation and m-imprecisiation are delicate tasks. The transformation of natural language terms into fuzzy sets and vice versa should be indeed “semantically nondestructive”, i.e. the intrinsic semantics of a linguistic term and the explicit semantics of the corresponding fuzzy set should be highly overlapping[2]. This relation between semantics is called “co-intension” and tightly constrains the precisiation processes [11].

Despite their symmetry in scope, m-precisiation and m-imprecisiation are asymmetrical in the processes carried out. Usually, the process of m-precisiation is not as difficult as the process of m-imprecisiation. For m-precisiation, a number of reference fuzzy sets are usually available, and linguistic terms are converted into fuzzy sets with

[2] We cannot guarantee identity in principle because the semantics of a linguistic term is subjective and represented within the synapses of the brain in a imperscrutable way.

a direct matching. Such reference fuzzy sets are manually defined or automatically designed through information granulation techniques.

On the other hand, m-imprecisiation is more delicate. The reason is immediate: linguistic terms are usually limited, whilst the class of fuzzy sets is much wider. If the information processing task does not take into account co-intension in its computation, the resulting fuzzy sets cannot be easily associated to any linguistic term. In this case m-imprecisiation requires further elaboration so as to extract a convenient natural language description of the results with eventual information loss.

Fuzzy information granules that can be associated to linguistic terms are called *interpretable*. Interpretability of fuzzy information granules requires a deep understanding of the semantical constraints involved in the user-computer interaction. Issues regarding the role of interpretability, its definition, evaluation and preservation need to be addressed. The rest of this Chapter is devoted in eliciting some of those interpretability issues which may come up when designing a HCIP system. The next sections give a formal definition of fuzzy information granule as well as an attempt to define interpretability in a very general sense. Then, a number of methodological issues are discussed regarding modeling with interpretable information granules. Interpretability constraints are then discussed and finally an outline of interpretable information granulation strategies is reported. The chapter ends with some concluding remarks highlighting open issues and future trends on this topic.

## 2 Fuzzy Information Granules

An information granule is a “clump of objects which are drawn together by indistinguishability, similarity, proximity or functionality” [9]. An information granule arises from a process of “granulation”, which refers to the mental act of dividing a whole into semantically significant parts.

The definition of information granule is open to several formalizations, e.g. intervals, rough sets, fuzzy sets, etc. [7]. In particular, a fuzzy information granule is defined through fuzzy sets on the domain of considered objects. Fuzzy Set Theory is hence the mathematical underpinning of the Theory of Fuzzy Information Granulation (TFIG). The difference in the two theories is primarily epistemic: FST is a pure mathematical theory, TFIG has more semantic concerns since the relation that keeps together objects in the same granule is fundamental for the definition of an information granule.

For complex domains (i.e. domains of objects characterized by several attributes), fuzzy information granules can be employed to represent pieces of knowledge. By assigning a name to each attribute, a fuzzy information granule can express a soft relationship between two or more attributes. A collection of fuzzy information granules can be used as a knowledge base to make inferences about the possible values of an attribute when the values of other attributes are given.

More formally, suppose that the object domain is $U = X \times Y$ (sub-domains *X* and *Y* could be multi-dimensional as well, but here we are not interested in the nature of such sets). A fuzzy information granule is completely represented by a fuzzy set on *U*, i.e. a fuzzy relation between objects in *X* and *Y*. We denote with Γ such a fuzzy set; we write $\Gamma : U \to [0,1]$ or, equivalently, $\Gamma : X \times Y \to [0,1]$. A knowledge base (also called “granular world” [7]) formed by fuzzy information granules $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$ is defined by accumulation of the pieces of knowledge defined by each single fuzzy information granule. This is a direct consequence of the definition of granulation, intended as a division of the whole (the domain *U*) into parts (the granules $\Gamma_i$ ).

The knowledge base (KB) is hence defined by the union of all the constituting fuzzy information granules, i.e.:

$$\mathrm{KB} = \bigcup_{i=1}^{n} \Gamma_i \tag{1}$$

To make inference, an information granule is interpreted as a possibility distribution over two variables *x* and *y*. When *x* is assigned a value (i.e. $x = \bar{x}$ ), inference is carried out by computing the possibility distribution of variable *y* for each information granule, namely $\pi_i^y$ , such that:

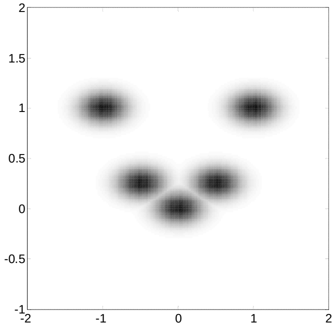
$$\pi_i^y(y) = \Gamma_i(\bar{x}, y) \tag{2}$$

A very common case is when each granule $\Gamma_i$ is defined as the Cartesian Product of two fuzzy sets, namely $A_i : X \to [0,1]$ and $B_i : Y \to [0,1]$ and $\Gamma_i(x, y) = A_i(x) * B_i(y)$ being ‘*’ a t-norm. In this case $\pi_i^y(y) = \Gamma_i(\bar{x}, y) = A_i(\bar{x}) * B_i(y)$. When the possibility distributions of all fuzzy information granules have been computed, the final possibility distribution for the entire KB is defined as $\pi^y(y) = \max_{i=1}^{n} \pi_i^y(y)$.

A KB made of fuzzy information granules is also called fuzzy graph [9]. Such structure is commonly used for approximating partially known functions, and is usually expressed in terms of a set of fuzzy rules, i.e. formal structures expressed as:

IF x is L[A] THEN y is L[B]

being L[A] and L[B] formal labels for fuzzy sets A and B respectively, and with the convention that “THEN” does not mean implication (as in logical rules) but it is a



*If x is about -1 then y is about 1*
*If x is about -½ then y is about ¼*
*If x is about zero then y is about 0*
*If x is about ½ then y is about ¼*
*If x is about 1 then y is about 1*

(a) (b)

**Fig. 2.** A granular parabola defined by five information granules: (a) graphical representation; (b) fuzzy rule representation

conjunction (it should be read “and then”). In fig. 2 a fuzzy graph representing a granular parabola is depicted.

Different interpretations of fuzzy rules are actually possible [12]. However, the interpretation of fuzzy rules as pieces of a fuzzy graph seems the most natural in the context of TFIG because it adheres to the definition of granulation and because the inference process is a direct application of the Compositional Rule of Inference.

We should note that the operator ‘*’ used for combining fuzzy sets into an information granule is usually a t-norm (the functional representation of the conjunction) and the aggregation of information granules into a knowledge base is achieved through a t-conorm (the functional representation of the disjunction). This choice is conformant with the definition of information granule and the process of granulation as a division of a whole into parts. Here the domain U is divided into information granules ($\Gamma_i$), hence the t-conorm acts as an aggregation operator that merges separate pieces of information.

On the other hand, the t-norm used for defining each $\Gamma_i = A * B$ is conformant with the definition of information granule, intended as a clump of objects kept together by a tying relationship. Indeed, according to the definition, an object (x,y) is in the granule $\Gamma_i$ if x is A AND y is B. If the t-norm is replaced by a t-conorm, the granule $\Gamma_i$ could be split into two granules $\Gamma_{1,i}$ and $\Gamma_{2,i}$ where an object (x,y) belongs to $\Gamma_{1,i}$ if x is A (and y is any), or (x,y) belongs to $\Gamma_{2,i}$ if y is B (and x is any). The need of a t-norm is a consequence of the use of a t-conorm for merging fuzzy information granules, which in turn is a consequence of the definition of the process of granulation as a division of a whole into parts.

With this in mind we should exclude logical rules (prolog like) as representing information granules, because these rules represent implication in the material form x is NOT A OR y is B. Actually information granules grasp the core of a rule, i.e. when both the antecedent and the consequent are valid. The availability of more information granules enables correct inference when the antecedent in an information granule is not verified. To validate inference, it is assumed that all the antecedents of rules cover the domain of input x.

Since each information granule grasps only the core of a rule, it is closer to the way human beings reason in terms of rules. It is well-known that material implication is sometimes counterintuitive, because of the validity of material implication when the antecedent is not verified (see, e.g. [13]). On the other hand, information granules represent pieces of knowledge that are more co-intensive with knowledge of users. In this sense, information granules seem a good form of knowledge representation for HCIP.

## 3 Interpretability of Fuzzy Information Granules

Information granules are the basic blocks for communicating information and knowledge from users to machines and vice versa. When communication starts from users and is directed to machines, information granules are built by a precisiation procedure, which is aimed at finding the most co-intensive representation of the perceptual information that is wished to be communicated to the machine. Several approaches have been proposed in literature, especially for fuzzy information granules. Here the key issue is the elicitation of the membership function of the fuzzy set representing the input (see, e.g. [14, Ch. 3]).

Different is the case when machines should communicate information granules. Such granules could result from deductive inference process, or could emerge from a process of inductive learning, e.g. after data clustering processes. In both cases the main problem is to give a representation of the information granule that could be easily understood by the recipient of the communication, i.e. it is co-intensive with some known concepts hold by the user.

When the information granule is a result of inference, usually defuzzification is applied, in order to reduce the granule to a single element of the domain (prototype). There is no doubt that this method is the most direct but the most wasteful. Information about the specificity (precision) of the information granule is lost, and usually no information is provided about the significance of the chosen prototype [15].

When information granules arise after a clustering process, the quest for interpretability becomes more stringent. Many clustering schemes exist in literature, which are able to find data clusters of several shapes. However, in the context of information granulation, the accuracy of the clustering process is only one of two facets: interpretability should be taken into account as well.

Interpretability is crucial in HCIP, especially when knowledge has to be extracted from data and represented in a comprehensible form. Interpretability is necessary to easily and reliably verify the acquired knowledge and to relate it to user's domain knowledge, to facilitate debugging and improving the granulation technique; to validate granules, for their maintenance, and for their evolution in view of changes in the external world [16-18]. This is especially important when the domain knowledge and the discovered knowledge must be merged together (e.g. in knowledge intensive systems) [19]. Finally, and maybe most importantly, interpretability is needed for convincing users that the model is reliable, especially when they are not concerned with the techniques underlying the granulation process. Users of a decision support system should be confident on how it arrives to its decisions. This is particularly important in domains such as medical diagnosis [20].

### 3.1 A Definition for Interpretability

Most interpretability-oriented modeling techniques adopt an interpretation of the "Occam's Razor" principle. The spirit of this approach is to guarantee interpretability by simplifying the description of the involved information granules. Several works on clustering go in this direction [21]. Whilst necessary, the accordance to the Occam's Razor principle misses the point of interpretability, i.e. co-intensivity with user knowledge, which is unquestionably a richer and more complex requirement than simplicity.

In order to get a deeper insight on interpretability, a suitable definition should be given to this quality. A suitable characterization for interpretability is given by the so-called "Comprehensibility[3] Postulate", proposed by Michalski, a scholar in the Machine Learning community [22]. The Comprehensibility Postulate states that:

[3] Throughout the chapter the terms "interpretability", "comprehensibility" and "understandability" are considered as synonyms.

> The results of computer induction should be symbolic descriptions of given entities, semantically and structurally similar to those a human expert might produce observing the same entities. Components of these descriptions should be comprehensible as single “chunks” of information, directly interpretable in natural language, and should relate quantitative and qualitative concepts in an integrated fashion.

The key point of the Comprehensibility Postulate is the human-centrality of the results of a computer induction process. According to the postulate the results of computer induction (e.g. information granulation) should be described symbolically. Symbols are necessary to communicate information and knowledge. Pure numerical methods, including neural networks, are hence not suited for meeting understandability unless an interpretability oriented post-processing of acquired knowledge is performed, such as in [16,23].

Symbolic descriptions are necessary but might not be sufficient. They should be structurally and semantically similar to those a human expert might produce observing the same entities. This means that highly complex mathematical relationships, though described symbolically, may not be interpretable because they may not be compatible with human cognition. In the same way, knowledge logically represented by a huge number of rules (or predicates, clauses, etc.) do not meet the understandability feature, since humans have a limited ability to store information in short-term memory [24]. This passage suggests simplicity as a necessary, albeit not sufficient, condition for interpretability.

According to the postulate, symbols (or structures of symbols) should represent chunks of information. Here we recognize information granules as the semantic counterparts of symbols used for communication. In order to be understandable, symbols should be directly interpretable in natural language. This does not necessarily mean that symbols should be chosen from a natural language vocabulary, but has more profound implications. In particular, the Comprehensibility Postulate requires the interpretation of symbols to be in natural language. This is a requirement on the semantics of the symbols, i.e. on the information granules they denote. Therefore, in order to be understandable, information granules should be conformed with concepts a user can conceive.

We further observe that natural language terms convey implicit semantics (which also depends on the context in which terms are used), that are shared among all human beings speaking that language. As a consequence, a symbol coming from natural language can be used to denote an information granule only if the implicit semantics of the symbol highly matches with the semantics characterized by the information granule.

Finally, as the Comprehensibility Postulate requires, the description of computer induction results should relate both qualitative and quantitative concepts in an integrated fashion. We recognize in this requirement the role of TFIG as the most suitable candidate for representing information [4]. Fuzzy information granules can indeed represent both quantitative information (e.g. through fuzzy numbers, fuzzy intervals, fuzzy vectors, etc.) and qualitative information (usually represented as adjectives such as “low”, “medium”, “high”, “etc”). Both types of information have a homogeneous representation and could be elaborated in an integrated fashion.

### 3.2 Interpretability and Natural Language

We should emphasize that the requirement of natural language descriptions becomes more apparent as the complexity of the knowledge to be communicated increases. This is in coherence with the well known Zadeh's incompatibility principle [25]. When the knowledge to be communicated is simple, natural language is not strictly necessary. For example, linear regression results or interval-based rules are easy to understand even though they are usually represented in a mathematical form. Actually, even these forms of representation are in coherence with the Comprehensibility Postulate.

For example a linear equation coming from regression is actually a prototype of a granule including all possible data distributions that can be approximated by the line, i.e. it represents a chunk of information. Furthermore, the simple structure of the linear equation can be directly described in natural language. For example, the following linear model:

$$y = 4.323432122x + 1.98726325$$

can be described as:

*y is proportional to x with factor about 4. For x=0 the value of y is about 2*

As another example, consider an interval-based rule, such as:

$$\text{IF } x \in [2.372138, 4.675121] \text{ THEN } y \in [0.061115, 1.512143]$$

Again, when trying to understand this rule, a user may not focus on its finest details. Rather, she would stop on a more abstract descriptive level that depends from the context. For example, the user would understand that:

*Whenever the value of x is about between 2.3 and 4.7, the value of y becomes smaller than about 1.5*

Even though natural language description is not necessary to communicate this forms of knowledge, high level concepts are actually formed in the user mind when trying to understand the results of computer induction processes. In all cases, what the user builds in her mind could be described in natural language. As the complexity of the model increases, any precise representation becomes less and less comprehensible. For high levels of complexity, natural language seems to be the only mean to communicate knowledge and to make it understandable by users.

### 3.3 Interpretability and Information Visualization

We note that often users understand problems if the information is properly visualized in some form. Indeed, visualization has been recognized as a viable mean to enable users to interpret large amounts of data and to gain deeper insight into the working of complex systems. Visualization has been extensively investigated to pursuit understanding of complex patterns or models. Recently, more attention has been devoted to develop approaches to visualize fuzzy data and fuzzy models. [26]. Some of these approaches have the primary objective of helping users in understanding how a model works to generate its behavior. Other visualization techniques are mainly aimed at graphically representing knowledge so that users could easily interpret them.

All such techniques may offer an invaluable help to users in understanding induction results, even if they may not involve the Comprehensibility Postulate as the final representation is not symbolic. We observe, however, that visualization techniques may not fulfill the understandability requirement of information granules. Indeed, they are very useful for understanding how a behavior is generated, but the user may not understand why such behavior is correct, in the sense of providing significant outcomes. Furthermore, the main reasons that justify the interpretability features may not be fulfilled by visualization tools. In this situation, visualization techniques are complementary to the Comprehensibility Postulate, rather than alternative.

# 4 Interpretability Issues

Interpretability of information granules is a complex requirement that needs a comprehensive analysis of all facets of the environment on which granules are developed and used. This analysis results in a number of issues to be addressed for fulfilling the interpretability requirement.

## 4.1 The Objective of Granular Model

A first issue to be addressed for interpretability is the objective of the granular model, which may have a twofold nature: descriptive and prescriptive.

When the scope of the model is to describe a phenomenon, a data structure, etc., a number of interpretability constraints should be adopted in order to meaningfully tag the information granules with linguistic labels (symbols). In many cases, however, the granules are also used to make inference for predictions concerning new data. Briefly speaking, information granules results become part of a prescriptive model in a decision support system.

In all situations where understandability is required, attention should be paid also on how predictions are inferred from information granules. Specifically, the inference process should be cognitively plausible, so as to convince users on the reliability of the derived decisions. This is a delicate step often left unaddressed.

As an example, let us consider a Mamdani Fuzzy Inference System (FIS) [27]. Mamdani rules are defined through fuzzy information granules, by separating input variables from output variables. Even though the rules embodied in these FIS are built by responding to all the interpretability requirements, the inference carried out by the system may not convince the users about the reliability of the derived decision. Usually, the output of a Mamdani FIS is attained by applying a defuzzification procedure to the inferred fuzzy set such as Center-of-Gravity (CoG). However, CoG may not have any plausible explanation in some domains (e.g. medical diagnosis). This type of defuzzification would not be convincing for the user (e.g. physicians) about the reliability of the inferred output. On the other hand, more plausible forms of defuzzification would provide for more plausible inferences [28].

### 4.2 Representation Structure

To achieve interpretability, the structure of knowledge plays a fundamental role. Roughly speaking, two classes of structures can be distinguished: single and multiple representation structures [29]. Single representation structures are based on a "flat" representation of the knowledge base, usually in a rule-based form. A different approach provides for a multiple representation of knowledge, where one representation (not necessary interpretable) is used to generate an accurate prescriptive model, while the other is used to describe knowledge in an interpretable form [30,31] Such dual representation has evidence in brain organization in which different areas are devoted to perception, action performing and natural language communication.

Information granules may offer valuable help in defining both single and multiple representation structures. Single representation structures are naturally derived by defining a rule for each information granule (as in fuzzy graphs), or by tying two information granules with an implicative connector. However, multiple representations are also possible by defining two or more levels of granulation of the same data. In this case, the top levels can be used for descriptive pursuits, while more accurate predictions could be taken through bottom level information granules, where interpretability is less stringent and hence information granules could take shapes more conformant to the underlying data structures. Techniques for context-based information granulation achieve multiple representation structures [33 Ch. 4, 34].

### 4.3 User Characterization

Interpretability concerns the characterization of the user accessing the knowledge base of a model. This characterization drives the choice of the most appropriate representation of information granules and, hence, the constraints required for granting their interpretability.

Users are mainly characterized by their needs. Some users might be interested in understanding the information granules derived from data in order to use them for their purposes. Other types of users could be more interested in the validity of information granules, especially in terms of predictive accuracy. For the second type of users, interpretability requirement is not as stringent as for users of the first type.

The verification of interpretability of information granules tightly constrains their shape. As a consequence, a strong bias is introduced in interpretable information granules, resulting in a weaker predictive accuracy w.r.t. information granules without interpretability requirements. The interpretability/accuracy tradeoff should be taken into account when designing a new system. The choice strongly depends on the need of the final users.

If users require understandability of information granules, then some guidelines should be followed. The domain expertise of the user helps in the choice of the most appropriate representation of information granules. For example, highly qualitative forms of representations might be useful for users who are primarily interested in understanding granulation results. On the other hand, more precise forms of representation would be more useful for users who make decisions on the basis of granulation results. Furthermore, the user vocabulary is helpful in the choice of the linguistic terms to be used for representing information granules. Finally, the required level of precision is important to choose the granularity level of information granules, as well as to decide for single or multiple levels of representation.

# 5 Interpretability Constraints

An important question concerns how to verify if an information granule is interpretable. The question is ill-posed because the definition of interpretability is blurry, subjective and context-dependent. However, a general approach can be set, which is mainly based on constraints.

Several interpretability constraints have been proposed in literature: a recent survey can be found in [35]. Some of them have a precise mathematical characterization, while others have a more fuzzy definition. This is expectable, since imprecise definitions of interpretability constraints may be more co-intensive with our perception of interpretability.

The choice of interpretability constraints mainly depends on user characterization, granulation objectives, etc. As already noted, any constraint imposed on information granules introduces new bias on their ability of representing data relationships. As a consequence, the choice of interpretability constraints should be as careful as possible.

Interpretability constraints can be organized in a hierarchy reflecting the level to which they are applied. A convenient hierarchy for fuzzy information granules is the following:

1. Constraints on one-dimensional fuzzy sets;
2. Constraints on frames of cognition[4];
3. Constraints on information granules;

In the following, a brief discussion on such constraints is reported. For a deeper argumentation, the reader is referred to [35].

## 5.1 Constraints on One-Dimensional Fuzzy Sets

In defining fuzzy information granules, we highlight their constitution as Cartesian products of one-dimensional fuzzy sets. This assumption helps to decompose information granules as conjunction of simpler properties, all characterized by one-dimensional fuzzy sets. Such fuzzy sets should be co-intensive with elementary concepts, usually represented in the form “v is A” being “v” the name of an attribute and “A” the name of a quality, whose semantics is defined by a fuzzy set.

In order to be co-intensive with elementary concepts, one-dimensional fuzzy sets should verify a number of constraints. The choice of such constraints is mostly driven by common-sense, as well as to avoid some paradoxical situations that can occur when they are violated.

### 5.1.1 Normality

All one-dimensional fuzzy sets should be normal, i.e. there exist atleast one element of the domain with full membership. Normality is required to avoid paradoxes such as

[4] A frame of cognition is intended as the set of all one-dimensional fuzzy sets defined for the same attribute.

inclusion in the empty set, as proved in [36]. However, this constraint is required for co-intensiveness since it is expected that qualities are always fully met by some elements of the domain. For unlimited domains, asymptotic normality is also acceptable (e.g. the concept “X is distant” could be represented in this way).

### 5.1.2 Convexity

A one-dimensional fuzzy set is convex if it is defined on an ordered domain (e.g. the real line). Convexity implies that any midpoint between two extremes has membership degree greater or equal to the minimum membership of the two extremes. Convex fuzzy sets are very common (e.g. triangular, trapezoidal, Gaussian, etc.) and they are widely used because they express the semantics of similarity-based qualities, i.e. all qualities for which sentences like “x is more A than y” make sense. Non-convex fuzzy sets can be used too, as proposed in [37], but usually they represent compound qualities, which should be denoted by complex linguistic labels (e.g. “mealtime”, meaning “breakfast or lunch or dinner”).

Strict convexity requires that the membership of any midpoint between two extreme points is strictly greater than the minimum membership of the two extremes. Gaussian fuzzy sets are strictly convex, while trapezoidal and triangular fuzzy sets are convex, but not strictly convex. Strictly convex fuzzy sets are suited for modeling concepts characterized by a magnitude. As an example, the concept “hot” is characterized by the perceived temperature. The degree to which a temperature is “hot” monotonically increases as the temperature increases. While a Gaussian fuzzy set effectively models such a relationship, a trapezoidal fuzzy set fixes a threshold after which any temperature is considered “hot” with the same degree. While justifiable for efficiency pursuits, the trapezoidal fuzzy sets may not as co-intensive as Gaussian fuzzy sets with the mental concept of “hot”, whose degree of evidence presumably varies with continuity.

### 5.1.3 Continuity

A fuzzy set defined on the real line should be continuous, so as to avoid abrupt changes of membership for very close elements. Continuity is especially required when fuzzy sets model perceptual information derived by observing macroscopic physical phenomena. Crisp set are examples of non-continuous fuzzy sets, leading to well known boundary paradoxes that make them unsuitable for modeling perceptual information and knowledge.

## 5.2 Constraints on Frames of Cognition

In any modeling context the number of fuzzy sets considered for each attribute is quite limited. The collection of all used fuzzy sets for a given attribute is named Frame of Cognition (FOC), as proposed in [36]. Linguistic Variables (defined in [38]) include a FOC if the number of generable linguistic values is finite. When a FOC is considered, a number of constraints should be verified in order to guarantee interpretability.

### 5.2.1 Proper Ordering

Linguistic terms used in natural language are related each other by two main relations: order and inclusion. As an example, the ordering of linguistic terms “small”, “medium” and “high” is quite intuitive. Also, in some contexts we understand that the semantics of “very L” is included in the semantics of the linguistic term L. As remarked in [39], interpretable fuzzy sets within a FOC should reflect such basic relationships.

Inclusion of fuzzy sets is straightforward: A is included in B iff the membership degree of any element to A is less than to B. Fuzzy sets defined on an ordered domain could be also partially ordered in the following way: given two fuzzy sets A and B, we say A<=B iff there exists a midpoint *t* such that each point less than *t* has membership to A greater than to B, and each point greater than *t* has membership to B greater than to A. Thus A better represents elements of the Universe of Discourse that are smaller than the elements represented by B. In this sense, the ordering of fuzzy sets reflects the semantics formalized by their membership functions. If this constraint is violated, undesired situations may occur, which hamper interpretability. As an example, given a FOC with two fuzzy sets “cold” and “hot”, we expect that for high temperatures the membership to hot is greater than the membership to cold, and vice versa for lower temperatures. This constraint calls for a proper choice of fuzzy sets in a FOC (see fig. 3 as illustration of an incorrect choice of fuzzy sets).

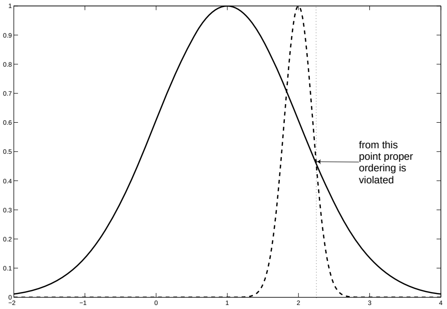


**Fig. 3.** Example of two Gaussian fuzzy sets violating proper ordering

### 5.2.2 Justifiable Number of Elements

In designing interpretable FOC, the number of fuzzy sets should be kept as small as possible, so that users could easily give appropriate meanings to the linguistic terms. By limiting the number of fuzzy sets in a FOC, a user is able to remember the proposed partition of the attribute domain. This greatly enhances interpretability.

The number of fuzzy sets is usually limited to 7±2, according to some psychological experiments reported in [24] showing the limited capacity of our short term memory in storing information. This limit has been debated (see [67] for a comprehensive discussion), and even smaller limits have been found in more recent experiments on immediate memory. Yet, a definitive answer on the capacity of primary memory has not been given, but psychological experiments (and common sense) suggest to keep the number of elements to remember very small.

The criterion of justifiable number of element spans all objects in a granular model. It is applicable to fuzzy sets in a FOC, as well as to the fuzzy sets compounding an information granule and to the number of information granules within a model. This extension is possible because our short-term memory is able to store simple structures (such as the names of fuzzy sets) as well complex structures (such as granule descriptions), provided that they are in small number.

We note that this criterion provides for a sound explanation of all simplification routines that are usually applied after clustering processes to enhance interpretability (see, e.g. [40] for a recent approach). However, we note also that this constraint severely limits the degrees of freedom (i.e. the free parameters) of a model. As a result, an interpretable model is highly biased and the resulting accuracy could be worse than an interpretability-free model. For this reason, interpretability is a feature that should be included with care in the design process.

### 5.2.3 Distinguishability

Roughly speaking, distinguishable fuzzy sets are well disjoint so they represent distinct concepts and can be assigned to semantically different linguistic labels. Well distinguishable fuzzy sets are deemed important since they obviate the subjective establishment of membership-function/linguistic term association, as claimed in [41], and reduce potential inconsistencies and redundancies in fuzzy models, as shown in [42]. Most importantly for the interpretability side, distinguishable fuzzy sets ease the linguistic interpretation of the model since fuzzy sets represent well separated concepts.

Distinguishability is a relation between fuzzy sets that can be formalized in several ways. Usually, a similarity measure between fuzzy sets is used, but the possibility measure can be also used under certain conditions, as showed in [43]. Possibility measure usually depends on the parameters of the membership functions, hence its calculation might be more efficient than similarity.

### 5.2.4 Coverage

The coverage constraint requires that every element of the domain belongs to at least one fuzzy set. Since membership is a matter of degree, the coverage constraint could be weak (membership greater than zero) or strong (membership greater than a threshold). In the latter case, the term $\alpha$-coverage is used, being $\alpha$ a threshold in $]0,1[$.

Coverage is related to completeness, a property of deductive systems that has been used in the context of Artificial Intelligence to indicate that the knowledge representation scheme can represent every entity within the intended domain [44]. In [45] coverage (there called “cover full range”) is justified by the fact that in human reasoning there will never be a gap of description within the range of the variable. On the contrary, as shown in [46] incompleteness may be a consequence of model adaption from data and can be considered a symptom of overfitting.

For the pursuits of interpretability, 0.5-coverage is desirable. This threshold corresponds to the optimal $\alpha$-cut when fuzzy sets are converted into crisp sets, as proved in [36]. This means that if 0.5-coverage is not guaranteed, then for some elements of the domain the FOC represent only negative qualities (e.g. x is not hot and not cold, see fig. 4). Usually, natural language has positive terms to describe such elements (e.g x is warm). As a consequence the inclusion of fuzzy sets in the FOC – so that 0.5-coverage is guaranteed – enhances the interpretability of the granular model.

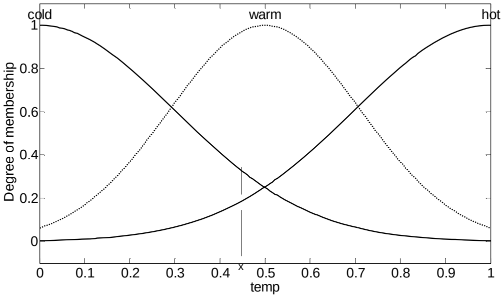


**Fig. 4.** Coverage and interpretability. Without fuzzy set labeled “warm”, values around 0.5 of the variable domain do not have a linguistic description but they can only be described as “not cold and not hot”.

### 5.2.5 Representation of Special Elements

For the pursuits of interpretability, it is often required that special elements of the universe of discourse are prototypes of some fuzzy sets in the FOC. In this way, such special elements are fully covered by some fuzzy sets which, in turn, represent special concepts.

Examples of special elements are extreme points of the universe of discourse. Leftmost and rightmost elements should be prototypes of some fuzzy sets that could be labeled in order to express their limit position in the FOC (e.g. “low”, “high”, “left”, “right”, etc.). The definition of such fuzzy sets is important to avoid paradoxical situations such as those depicted in fig. 5a. If the three fuzzy sets represent concepts “small”, “medium”, “high” then the leftmost element minU is less “small” than minU+e. Such an undesired situation will not occur if leftmost/rightmost fuzzy sets are defined in the FOC (fig. 5b). Leftmost and rightmost fuzzy sets are necessary when they represent qualitative concepts; if the FOC is made of fuzzy sets expressing fuzzy quantities, then they are not necessary.

Other special elements could be considered as prototypes of some fuzzy sets. The choice of such elements is problem driven. As an example, in [44] it is suggested that for control applications the null value, if it belongs to the universe of discourse,

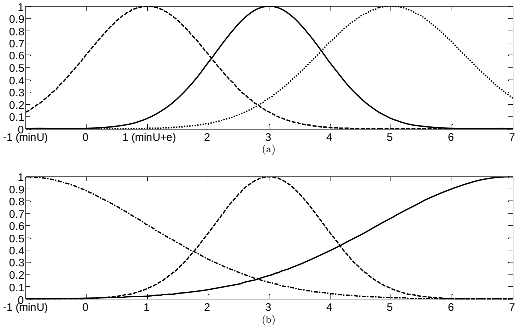


**Fig. 5.** (a) A FOC violating representativity of extreme values. (b) a FOC with leftmost/rightmost fuzzy sets.

should be prototype of a fuzzy set, expressing the concepts “nearly zero”. This suggestion could be extended to other values deemed important in an applicative context. For example, 0 and 100 could be prototypes of fuzzy sets labeled “icing point” and “boiling point” in the domain of water temperatures (expressed in Celsius degrees), or 37°C could be considered as prototype in a FOC expressing human body temperatures.

## 5.3 Interpretability Constraints on Information Granules

A fuzzy information granule is defined as a Cartesian product of fuzzy sets, each coming from a different FOC. The relational nature of information granules make them the basic building blocks for expressing knowledge. To make such knowledge comprehensible, a number of interpretability constraints should be verified.

### 5.3.1 Justifiable Number of Elements

The constraint of justifiable number of elements (JNE in brief) is applied at all levels of granulation: the FOC, each single information granules and the collection of information granules in a model.

When considering a single information granule, the JNE constraint imposes that the number of fuzzy sets defining the granule should be kept small, e.g. less than about seven. In this way, it is easier for the user to build a mental concept associated to the granule. Very complex granules are difficult to understand and, even if the compounding fuzzy sets verify all suggested interpretability constraints, users may not be able to grasp the relationship among variables that is represented by the granule. Techniques such as variable selection, locally to the granule or globally to the entire knowledge base, are useful to improve interpretability.

When considering the overall knowledge base, the JNE constraint suggests that the number of information granules should be kept within a small limit. According to psychological experiments, indeed, our short term memory is able to store simple as well as complex structures, provided that they are in a small number. Again, this poses severe limits on the flexibility of the resulting model, which may negatively influence its accuracy.

### 5.3.2 Completeness

This constraint imposes that each element of the universe of discourse is covered by at least one information granule, i.e. it belongs to an information granule with a membership degree greater than zero (weak completeness) or greater than a specified threshold (strong completeness).

Weak completeness is easy to achieve if fuzzy sets with infinite support are used (such as Gaussian fuzzy sets). On the other hand, strong completeness may pose some problems, especially when the specified threshold is high (e.g. 0.5 as usually required) and the dimensionality of the universe of discourse is high. In this case, indeed, a great number of information granules may be required, which may hamper the constraint of justifiable number of elements. To overcome this problem a form of “closed world assumption” is usually adopted. It is assumed that all elements that can occur are represented by at least one information granule with membership degree greater than a threshold. According to this assumption, the universe of discourse is actually a subset of the Cartesian product of all attribute domains, and it is assumed that this subset is covered by the information granules.

A drawback of the closed world assumption emerges when an element not belonging to the subset occurs. In this case the model may infer weak results (i.e. highly subnormal fuzzy sets). To overcome this problem, in [47] a "default" information granule is used, whose membership function is defined as the complement of the union of all used information granules. In this way, any element not represented by any information granule is covered by this default granule, to which a special action could be attached.

### 5.3.3 Correctness

The correctness constraint applies to the inference process carried out by the granular model. Informally speaking, correctness imposes that the inference process provides logically consistent outputs. As an example, in rule-based models, correctness requires that Modus Ponens is respected, i.e. if a rule of the type "IF x is A THEN y is B" belongs to the rule base, and the input A is provided, it is expected that the model output is B.

On the basis of this definition, several efforts have been made to verify the correctness of rule-based models, such as in [48]. For granular models, we should keep in mind that the knowledge base is made by the union of information granules, and each information granule is defined by the conjunction of elementary concepts. As a consequence, if a granule representing "x is A and y is B" belongs to the model, and the input A is provided, then the model is expected to derive B possibly united with other fuzzy sets.

As an example, consider a model with two information granules, labeled as "temperature is very cold and position is north pole", and "temperature is very cold and position is south pole". If the fact "temperature is very cold" is provided, we should expect the inference "position is north pole OR south pole". In this sense, rule inconsistencies are not possible in granular models (see also [49] for a formal treatment of the topic).

## 6 Interpretable Fuzzy Information Granulation

Information granulation is the process of discovering granules from data by extracting hidden relationships among observed samples. The nature of such relationships depends on the granulation algorithm and defines the semantics of the resulting information granules.

According to Zadeh [4], granulation is a cognitive task devoted to the partition of a whole into (significant) parts. Conversely, the process of aggregating parts into a whole is referred as organization. By virtue of such a definition, we may interpret granulation as the discovery of relationships of data *within* parts, so that the latter are semantically significant. On the other hand, organization involves the discovery of relationships *between* parts.

Interpretable information granulation adds interpretability constraints to the granulation process. The choice of which constraints to include is a matter of design. In the following, a number of commonly adopted strategies for interpretable information granulation is outlined. For a deeper review of interpretable granulation techniques (in the context of fuzzy modeling) the reader is referred to [50].

### 6.1 Partitioning

A widely adopted strategy for interpretable information granulation concerns the partition of the data domain into fuzzy granules that verify a number of interpretability constraints.

Partitioning can be fixed or dynamic. In fixed partitioning, a number of interpretable fuzzy sets is defined for each attribute (a FOC), and information granules are obtained by combining fuzzy sets of different attributes. To avoid combinatorial explosion of information granules, only those including an adequate number of available samples is retained, while all the others are discarded.

Fixed partitioning provides for very interpretable fuzzy information granules but suffers of many drawbacks. The main shortcoming derives from the definition of fuzzy sets, which does not take into account the structure of data. As a consequence fixed partitioning may not represent the most adequate granulation of data. Furthermore, the choice of the number of fuzzy sets for each attribute determines the granularity level of each information granule. Without any information of data distribution, an arbitrary choice of the level of granularity may seriously hamper the quality of the granulation process. Despite these drawbacks, however, manual partitioning is still widely used because of its simplicity [63].

To avoid the shortcomings of fixed partitioning, dynamic partitioning techniques have been proposed. The fundamental strategy of dynamic partitioning is to refine an initial partition so as to better represent data relationships, without violating interpretability constraints. Refinement usually applies merge and split operators for fuzzy sets [51, 64], or modification of fuzzy set parameters [52], or both [53]. In [65] fuzzy sets in a FOC are defined by a frequentist approach so that more specific fuzzy sets are defined to cover attribute values with higher frequency.

Alternative to these partition strategies, some works use fuzzy tree-based partitioning to granulate data [54]. Roughly speaking, for generating a tree-based partition an attribute is selected and split in two fuzzy sets. For each of the two parts the split algorithm is applied on the remaining attributes. This approach leads to a compact representation of information granules (especially because attribute selection is usually performed), but the resulting granules may not share fuzzy sets. This implies a number of similar fuzzy sets to be defined for the same attribute, which may hamper interpretability (fig. 6).

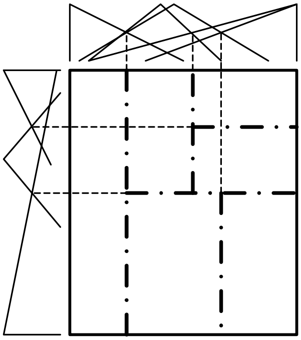


**Fig. 6.** Example of tree-based partition

### 6.2 Clustering

Clustering techniques are widely used for data granulation. This is due to the ability of clustering techniques to discover hidden relationships from data. Several fuzzy clustering techniques have been proposed in literature (see [55] for a review), however few of them address interpretability.

The main difficulty for assuring interpretability of fuzzy granules resulting from clustering processes generally stands in the difficulty of representing fuzzy clusters in natural language. A common approach to assure natural language representation is to express fuzzy clusters as Cartesian product of fuzzy sets. This however implies that the shape of fuzzy clusters is tightly constrained. Furthermore, the presence of several clusters may lead to a high number of very overlapped fuzzy sets for each attributes. This situation does not lead to an interpretable granulation of data.

To improve interpretability of cluster-based information granules, often simplification procedures are proposed, which merge similar fuzzy sets of the same FOC into single fuzzy sets [56]. This approach yields compact granular models, but often other interpretability constraints (e.g. coverage, representativity, etc.) are not fulfilled. They are hence most suited for quantitative information granulation, where granules represent imprecise quantities for each attribute.

When qualitative information granulation is required, i.e. when granules represent qualities on each attributes, a greater number of interpretability constraints should be verified. In this case, other strategies are advisable. In [57] an approach for interpretable granulation is proposed, which is based on a double clustering process. The first clustering stage operates on the entire dataset in order to discover hidden relationships among data. The result of this stage is a collection of prototypes that synthetically describe the dataset. In the second stage, multidimensional prototypes are projected onto each single attribute and further clustered to achieve the desired granulation level. One-dimensional projections are used to define FOCs that verify a number of interpretability constraints so that the resulting fuzzy sets can be labeled with qualitative linguistic terms. Such fuzzy sets are combined (one for each attribute) in order to define fuzzy information granules that represent data in a natural language form. Variants of the Double Clustering schema are also able to automatically determine the granularity level [58,59].

## 7 Concluding Remarks

The main objective of interpretability in fuzzy information granulation is co-intensiveness with human perceptual knowledge. Fuzzy information granules are basic building blocks for representing semantical knowledge in a computer-manageable form. Without interpretability, fuzzy information granules are still able to represent imprecise knowledge, similarly to the knowledge learned by a neural network or some other black box model, but its comprehensibility by users is limited, especially when they do not possess skills in fuzzy and granular technologies.

In this sense, interpretability is a fundamental feature for using fuzzy information granules in Human-Centric Information Processing. But interpretability is not a mathematical property, it is rather an epistemic feature that spans several facets of model design. Interpretability constraints are the formal counterparts of the interpretability property, provided that several other issues have been addressed, such as user characterization, representation structure and, last but not least, the objective of the model to be designed.

Current literature offers a great number of interpretability constraints, some of which have been revised in this chapter. Many of these constraints have been proposed as formalizations of properties driven by common sense. This opens the door to further research aimed at finding relationships (e.g. the “representation of special elements”, which is a generalization of two or more constraints found in literature), at devising different formalizations of the same constraint (such as the “distinguishability” constraint or the “proper ordering”) or at discarding some proposed constraint (such as the 1-complementarity of membership degrees, which has a technical rationale but cannot be justified in terms of interpretability). Even more importantly, groups of interpretability constraints help in identifying different notions of interpretability, such as interpretability of granules expressing quantities rather than qualities. These and other findings are important aids for knowledge engineering with fuzzy information granules.

Interpretability usually clashes with predictive accuracy. The more interpretability constraints are used, the more rigid is the granular model, and the less flexible to adaption it is. Interpretability vs. accuracy tradeoff has been addressed for long time, and several approaches have been proposed to balance these two features e.g. by regularized learning or multi-objective optimization (see, e.g. [54, 60]). Furthermore, the adoption of interpretability constraints should be carefully pondered in certain applications where accuracy has a prominent importance, such as in fuzzy control [61].

Current and future research on interpretability spans both methodological and theoretical issues. Among these, the representation of the semantics of natural language terms is of particular interest. Mendel [62] proposes type-2 fuzzy sets for such a representation. This is a promising research direction, which may result particularly fruitful in the area of granular knowledge communication (see also [66] for a discussion on this topic). On a more general level, we believe that deep insights on the semantics of membership degrees (which could denote similarity, preference, possibility of other, see [68]), as well as their operations, will shed new light on interpretability of information granules and, in turn, on Human-Centered Information Processing.

## References

1. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering. Toward Human-Centric Computing. John Wiley & Sons, Hoboken (2007)
2. Pazzani, M.J.: Knowledge discovery from data? IEEE Intell. Syst. 15(2), 10–12 (2000)
3. Jaimes, A., Gatica-Perez, D., Sebe, N., Huang, T.S.: Human-Centered Computing–Toward a Human Revolution. IEEE Computer 40(5), 30–34 (2007)
4. Zadeh, L.A.: Toward a Theory of Fuzzy Information Granulation and its Centrality in Human Reasoning and Fuzzy Logic. Fuzzy Sets and Syst. 90, 111–117 (1997)
5. Zadeh, L.A.: Graduation and granulation are keys to computation with information described in natural language. In: Proc. of IEEE GrC 2006, p. 30 (2006)
6. Zadeh, L.A.: Toward Human-Level Machine Intelligence. In: Proc. of SOFA 2007, pp. 9–10 (2007)

7. Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Kluwer Academic Publishers, Dordrecht (2003)
8. Bargiela, A., Pedrycz, W.: Toward a Theory of Granular Computing for Human-Centered Information Processing. IEEE Trans. on Fuzzy Syst. 16(2), 320–330 (2008)
9. Zadeh, L.A.: Toward a generalized theory of uncertainty (GTU): an outline. Inf. Sci. 172, 1–2 (2005)
10. Zadeh, L.A.: Is there a need for fuzzy logic? Inf. Sci. 178(13), 2751–2779 (2008)
11. Zadeh, L.A.: A New Frontier in Computation – Computation with Information Described in Natural Language. In: Proc. of 2006 IEEE Conf. on Intell, pp. 5–6 (2006)
12. Dubois, D., Prade, H.: What are fuzzy rules and how to use them. Fuzzy Sets Syst 84(2), 169–185 (1996)
13. Brandom, R.: Semantic Paradox of Material Implication. Notre Dame J. of Form. Log. 22(2), 129–132 (1981)
14. Türksen, I.B.: An Ontological and Epistemological Perspective of Fuzzy Set Theory. Elsevier, Amsterdam (2005)
15. Roychowdhury, S.: An Inquiry into the Theory of Defuzzification. In: Pedrycz, W. (ed.) Granular Computing: an Emerging Paradigm, pp. 143–162. Physica-Verlag (2001)
16. Zhou, Z.H.: Comprehensibility of Data Mining Algorithms. In: Wang, J. (ed.) Encyclopedia of Data Warehousing and Mining, Hershey, PA, pp. 190–195 (2005)
17. Giboin, A.: ML Comprehensibility and KBS explanation: stating the problem collaboratively. In: Proc. of IJCAI 1995 Workshop on Machine Learning and Comprehensibility, pp. 1–11 (1995)
18. Mencar, C., Castellano, G., Fanelli, A.M.: On the role of interpretability in fuzzy data mining. Int. J. of Uncertain., Fuzziness and Knowl. Based Syst. 15, 521–537 (2007)
19. Michalski, R.S.: Knowledge Mining: A Proposed New Direction. In: Proc. of Symp. on Data Mining and Semantic Web (2003), `http://www.mli.gmu.edu/papers/2003-2004/03-5.pdf`
20. Kuncheva, L.I., Steimann, F.: Fuzzy diagnosis. Art. Intell. in Medicine 16(2), 121–128 (1999)
21. Guillaim, S.: Designing fuzzy inference systems from data: An interpretability-oriented review. IEEE Trans. Fuzzy Syst. 9(3), 426–443 (2001)
22. Michalski, R.S.: A theory and methodology of inductive learning. Artif. Intell. 20, 111–161 (1983)
23. Johansson, U., Niklasson, L., Knig, R.: Accuracy vs. Comprehensibility in Data Mining Models. In: Proc. of the 7th Int. Conf. on Inform. Fusion, pp. 295–300 (2004)
24. Miller, G.A.: The magical number seven, plus or minus two: Some limits on our capacity for processing information. The Psychological Review 63, 81–97 (1956), `http://www.well.com/user/smalin/miller.html` (electronic reproduction by Stephen Malinowski)
25. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on Syst., Man and Cybern. 3(1), 28–44 (1973)
26. Pham, B., Brown, R.: Visualisation of fuzzy systems: requirements, techniques and framework. Future Gener. Comput. Syst 21(7), 1199–1212 (2005)
27. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a Fuzzy Logic controller. Int. J. of Human-Comput. Studies 51(2), 135–147 (1999)
28. Roychowdhury, R., Pedrycz, W.: A survey of defuzzification strategies. Int. J.of Intell. Syst. 16(6), 679–695 (2001)
29. van de Merckt, T., Decaestecker, C.: Multiple- Knowledge Representations in Concept Learning. In: Lavrač, N., Wrobel, S. (eds.) ECML 1995. LNCS (LNAI), vol. 912, pp. 200–217. Springer, Heidelberg (1995)

30. Domingos, P.: Knowledge Discovery Via Multiple Models. Intell. Data Analysis 2(3), 187–202 (1998)
31. Sun, R., Slusarz, P., Terry, C.: The Interaction of the Explicit and the Implicit in Skill Learning: A Dual-Process Approach. Psych. Rev. 112(1), 159–192
32. Stadler, M., Frensch, P.: Handbook of implicit learning. Sage, Thousand Oaks (1998)
33. Pedrycz, W.: Knowledge Based Clustering: from Data to Information Granules. John Wiley & Sons, Hoboken (2005)
34. Mencar, C., Castellano, G., Fanelli, A.M.: Balancing Interpretability and Accuracy by Multi-Level Fuzzy Information Granulation. In: Proc. of IEEE Int. Conf. on Fuzzy Syst., pp. 2157–2163 (2006)
35. Mencar, C., Fanelli, A.M.: Interpretability constraints for fuzzy information granulation. Inf. Sci. 178(24), 4585–4618 (2008)
36. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets: Analysis and Design. MIT Press, Cambridge (1998)
37. Garibaldi, J.M., Musikasuwan, S., Ozen, T., John, R.I.: A Case Study to Illustrate the Use of Non-Convex Membership Functions for Linguistic Terms. In: Proc. of IEEE Int. Conf. on Fuzzy Syst., vol. 3, pp. 1403–1408 (2004)
38. Zadeh, L.A.: The concept of Linguistic Variable and its application to approximate reasoning. Inf. Sci. 8, 199–249 (1975)
39. Bodenhofer, U., Bauer, P.: Interpretability of linguistic variables: a formal account. Kybernetika 41(2), 227–248 (2005)
40. Gegov, A.: Complexity Management in Fuzzy Systems: A Rule Base Compression Approach. Studies in Fuzziness and Soft Computing, vol. 211. Springer, Heidelberg (2007)
41. de Oliveira, J.V.: Semantic constraints for membership function optimization. IEEE Trans. on Syst., Man and Cybern. A 29(1), 128–138 (1999)
42. Roubos, H., Setnes, M.: Compact fuzzy models through complexity reduction and evolutionary optimization. In: Proc. of 9th IEEE Int. Conf. on Fuzzy Syst., vol. 2, pp. 762–767 (2000)
43. Mencar, C., Castellano, G., Fanelli, A.M.: Distinguishability quantification of fuzzy sets. Inform. Sci. 177, 30–149 (2007)
44. de Oliveira, J.V.: Towards neuro-linguistic modeling: Constraints for optimization of membership functions. Fuzzy Sets and Syst. 106, 357–380 (1999)
45. Hermann, C.S.: Symbolic reasoning about numerical data: A hybrid approach. Appl. Intell. 7, 339–354 (1997)
46. Jin, Y., Von Seelen, W., Sendhoff, B.: On generating FC3 fuzzy rule systems from data using evolution strategies. IEEE Trans. on Syst., Man and Cybern. B 29(6), 829–845 (1999)
47. Peña-Reyes, C., Sipper, M.: Fuzzy CoCo: Balancing Accuracy and Interpretability of Fuzzy Models by Means of Coevolution. In: Casillas, J., Cordon, O., Herrera, F., Magdalena, L. (eds.) Accuracy Improvements in Linguistic Fuzzy Modeling, pp. 119–146. Springer, Heidelberg (2003)
48. Perfilieva, I., Lehmke, S.: Correct models of fuzzy IF–THEN rules are continuous. Fuzzy Sets and Syst. 157(24), 3188–3197 (2006)
49. Dubois, D., Prade, H., Ughetto, L.: Checking the Coherence and Redundancy of Fuzzy Knowledge Bases. IEEE Trans. on Fuzzy Syst. 5(3), 398–417 (1997)
50. Guillaume, S.: Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. IEEE Trans. on Fuzzy Syst. 9(3), 426–443 (2001)
51. Guillaume, S., Charnomordic, B.: Generating an Interpretable Family of Fuzzy Partitions From Data. IEEE Trans. on Fuzzy Syst. 12(3), 324–335 (2004)

52. de Oliveira, J.V.: Towards neuro-linguistic modeling: Constraints for optimization of membership functions. Fuzzy Sets and Syst. 106(3), 357–380 (1999)
53. Nauck, D.D.: Fuzzy data analysis with NEFCLASS. Int. J. on Approx. Reason. 32(2-3), 103–130 (2003)
54. Mikut, R., Jäkel, J., Gröll, L.: Interpretability issues in data-based learning of fuzzy systems. Fuzzy Sets and Syst. 150(2), 179–197 (2005)
55. Baraldi, A., Blonda, P.: A survey of fuzzy clustering algorithms for pattern recognition. Parts I & II. IEEE Trans. on Syst., Man, and Cybern. B 29(6), 778–801 (1999)
56. Roubos, H., Setnes, M.: Compact and transparent fuzzy models and classifiers through iterative complexity reduction. IEEE Trans. on Fuzzy Syst. 9(4), 516–524 (2001)
57. Castellano, G., Fanelli, A.M., Mencar, C.: DCf: a double clustering framework for fuzzy information granulation. In: Proc. of IEEE Intern. Conf. on Granular Comput., vol. 2, pp. 397–400 (2005)
58. Castellano, G., Fanelli, A.M., Mencar, C., Plantamura, V.: Classifying data with interpretable fuzzy granulation. In: Proc. of SCIS&ISIS, pp. 872–877 (2006)
59. Mencar, C., Consiglio, A., Fanelli, A.M.: DC$\gamma$: Interpretable Granulation of Data through GA-based Double Clustering. In: Proc. of IEEE Int. Conf. on Fuzzy Syst., pp. 1–6 (2007)
60. Ishibuchi, H., Nojima, Y.: Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. Int. J. of Approx. Reason. 44, 4–31 (2007)
61. Riid, A., Rustern, E.: Interpretability of Fuzzy Systems and Its Application to Process Control. In: Proc. of IEEE Int. Conf. on Fuzzy Syst., pp. 1–6 (2007)
62. Mendel, J.M.: Computing with words and its relationships with fuzzistics. Inform. Sci. 177, 988–1006 (2007)
63. Novák, V., Perfilieva, I., Dvořǎk, A., Chen, G., Wei, Q., Yan, P.: Mining pure linguistic associations from numerical data. Int. J. on Approx. Reason. 48, 4–22 (2008)
64. Herrera, F., Herrera-Viedma, E., Martínez, L.: A Fuzzy Linguistic Methodology to Deal With Unbalanced Linguistic Term Sets. IEEE Trans. on Fuzzy Syst. 16(2), 354–370 (2008)
65. Drobics, M., Himmelbauer, J.: Creating comprehensible regression models. Soft. Comput. 11, 421–438 (2007)
66. Dimitrov, V., Russell, D.: The Fuzziness of Communication. Fell L, Russell D, Stewart A, Seized by Agreement, Swamped by Understanding. Hawkesbury Printing, University of Western Sydney. (1994), `http://www.univie.ac.at/constructivism/pub/seized/fuzcom.html`
67. Cowan, N., Morey, C.C., Chen, Z.: The legend of the magical number seven. In: Della Sala, S. (ed.) Tall Tales about the Mind and Brain - Separating Fact from Fiction, pp. 45–59. Oxford University Press, Oxford (2007)
68. Dubois, D., Prade, H.: The three semantics of fuzzy sets. Fuzzy Sets and Sys 90(2), 141–150 (1997)

# Semantic Driven Fuzzy Clustering for Human-Centric Information Processing Applications

Paulo Fazendeiro[1,2] and José Valente de Oliveira[2]

[1] University of Beira Interior, Department of Informatics, Av. Marques d'Ávila e Bolama, 6200-001 Covilha, Portugal
fazendeiro@ubi.pt

[2] University of Algarve, UAlg Informatics Lab, Campus de Gambelas, 8000-062 Faro, Portugal
jvo@ualg.pt

**Abstract.** This chapter presents an overview of fuzzy clustering techniques aiming at human-centric information processing applications and introduces the accuracy-interpretability tradeoff into the conceptualization of the clustering process. Nowadays it is a matter of common agreement that the cornerstone notion of information granulation is fundamental for a successful outcome of exploratory data analysis and modeling in fields like science, engineering, economics, medicine and many others. There is no doubt that fuzzy clustering is an excellent medium to obtain such information granules. For a matter of self-containment the chapter starts by presenting the fundamentals of fuzzy clustering along with some variants and extensions. In the second part of the chapter, the fuzzy clustering approach is highlighted as a valuable human-centric interface: the roadmap from data to information granules is displayed along with a discussion on some mechanisms to implement user relevance feedback. In the last part of the chapter a semantic driven evolutionary fuzzy clustering algorithm is analyzed, as a particular instance of a class of unsupervised clustering algorithms which embraces constraints usually applied in supervised learning. The results show that these more general constraints while tuning the equilibrium between accuracy and interpretability concomitantly help to unveil the structural information of the data.

## 1 Introduction

From the strictly conceptual point of view the human-centric development paradigm and the human-centric information processing pursuits share the final goal of making the synthesized system highly adaptable to the needs of the human user and capable of presenting a natural interface which fosters the user-system interaction. In this sense fuzzy clustering techniques undoubtedly have an important place in the human-centric armamentarium. The contribution of fuzzy clustering techniques to the human-centric paradigm effort can be roughly enumerated along three different application vectors. First of all, from the end-user perspective, the intuitive nature of the fuzzy sets offers a privileged mean of communication of the exploratory data analysis findings in a user-friendly way.

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 119–151.
springerlink.com

Sections 3.1 and 3.2 present a detailed discussion of this aspect. The second facet, perhaps more interesting from the system designer's point of view but also very important to the user interaction and feedback to the system, has to do with the seamless integration of human knowledge as a support or guiding mechanism of the clustering activity. In Sect. 3.3 we pinpoint some interesting trends in this regard. Finally a third aspect, less obvious and more related to the behavior of the system being developed, concerns the incorporation of human-defined semantic interpretability constraints into the clustering process and taking advantage of these as a mean to accomplish a transparent system with good accuracy. In Sect. 4 we present such constraints along with their generalizations to the clustering framework and report some results which point to the merit of considering them from inception.

The chapter is organized into three main parts. Section 2 presents an overview of fuzzy clustering techniques emphasizing the competitive advantage over crisp clustering, pointing problems, shortcomings and possible solutions. Section 3 discusses the development of information granules and mechanisms of user customization. Whereas these two parts highlight the role of the fuzzy clustering algorithmic framework as support to the human-centric paradigm Sect. 4 sustains that designing the clustering algorithm in a human-centric way can be a valuable asset. Following this perspective, simple human-centric semantic constructs that are commonly used in supervised learning as a way to balance interpretability-accuracy equilibrium are successfully transposed to unsupervised fuzzy clustering, resulting in clustering techniques which provide interesting results from the end-user point of view.

## 2 Overview of Fuzzy Clustering

Generally speaking clustering is the process of searching for a finite and discrete set of data structures (categories or clusters) within a finite, otherwise unlabeled, usually multi-variate data set. In the literature it is common to find that the goal of clustering is the partition of the data set into groups so that data in one group are similar to each other and are as different as possible from data in other groups, cf. [42, 75]. Two complementary facets are enclosed in this unsupervised learning task: the elicitation of a model of the overall structure of the data and the pursuit for a manageable representation of a collection of objects into homogeneous groups.

The notion of similarity between elements (patterns) of the data set is a concept of paramount importance, with direct implications on the clustering endeavor. Usually the similarity is defined at the expense of some appropriate distance function. In cluster analysis common choices for distance functions include the Hamming (city block) distance inducing diamond shaped clusters, the Euclidean distance inducing (hyper) spherical clusters and the Tchebyshev distance inducing (hyper) box shaped clusters. As a matter of fact these examples are members of the Minkowski family of distances, or $L_p$ norms, defined as:

$$D(\mathbf{x},\mathbf{y}) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} . \tag{1}$$

Distance can be used to measure the similarity between two data points or between a data point and a prototype of the cluster. The prototype is a mathematical object, usually a point in the feature space (e.g. the center of the cluster) or even a geometric subspace or function, acting as a representative of the cluster while trying to capture the structure (distribution) of its associated data.

Traditionally the clustering algorithms are categorized in two main types: hierarchical and objective function based partitional clustering. Every new cluster determined by a hierarchical algorithm is based on the set of previously established clusters. The distance between individual points has to be generalized to the distance between subsets (linkage metric) in order to merge (or split) clusters instead of individual points. The type of the used linkage metric significantly affects hierarchical algorithms, since each cluster may contain many data points and present different geometrical shapes, sizes and densities. The distance is computed for every pair of points with one point in the first set and another point in the second set. Due to the pairwise combinatorial nature of the process the hierarchical approach tends to be computationally inefficient with the growth of the number of data elements. This approach is very sensitive to anomalous data points (noise and outliers) and is unable to handle overlapping clusters. A reason for this is that bad decisions made at an early stage of the algorithm will be propagated and amplified up to the end since the intermediate clusters are not revisited for further improvement (the points can not move from one cluster to another).

The second major category of clustering algorithms attempts to directly decompose the data set into a collection of disjoint clusters. This partition is builded during an iterative optimization process repeated until its associated cost function reaches a minimum (global or local). The cost function, also designed performance index or objective function, is a mathematical criterion expressing some desired features (emphasizing local or global structure of the data) of the resulting partition.

Combining some heuristics with an adequate formulation of the objective function it is possible to design an optimization process which is able to determine at least suboptimal partitions. One such formulation, for that matter the most used in practice, is the sum-of-squared-error distances or minimum variance criterion representing each of C clusters by their mean (the so-called centroid $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, \ldots, C$ of its points):

$$Q = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij} D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i) \tag{2}$$

where $\mathbf{X}= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ denotes a set of feature vectors (or patterns) in the $\mathbb{R}^n$ space. $D_{ji}(\mathbf{x}_j, \mathbf{v}_i)$ is a measure of the distance from $\mathbf{x}_j$ to the $i$th cluster prototype. The elements, $u_{ij} \in \{0, 1\}$, $i = 1, \ldots, C$ $j = 1, \ldots, N$ form a matrix designated as the *partition matrix* which maps the patterns to a cluster. If

$u_{ij} = 1$ the pattern $j$ belongs to cluster $i$, otherwise if $u_{ij} = 0$ the pattern $j$ is not accounted as a member of cluster $i$. This formulation is appealing because it still favors sets of well separated clusters with small intra-cluster distances whereas replaces all the pair-wise distances computation by a single cluster representative. Thus the computation of the objective function becomes linear in $N$ and it is now feasible the application of an iterative optimization process aiming at gradual improvements of the builded clusters.

The $c$-Means algorithm (also referred in the literature as $k$-Means or hard $c$-Means) [26,54] is the best known squared error-based example of such a process. For a given initialization of the $C$ centroids the heuristic approach consists of two-step major iterations that follow from the first-order optimality conditions of (2): first reassign all the points to their nearest cluster, thus updating the partition matrix, and then recompute the centroids, $\mathbf{v}_j$ (its coordinates are the arithmetic mean, separately for each dimension, over all the points in the cluster), of the newly assembled groups. This iterative procedure continues until a stopping criterion is achieved (usually until no reassignments happen). In spite of its simplicity and speed this algorithm has some major drawbacks. It is much dependent on the initial centroids assignment (frequently in practice it is run for a number of times with different random assignments and the best resulting partition is taken), does not ensure that the result has a global minimum of variance, is very sensitive to outliers and lacks scalability.

Another not so obvious disadvantage is related to the binary nature of the elements of the partition matrix and consequently of the induced partitions. This kind of partition matrix is based on classical set theory, requiring that an object either does or does not belong to a cluster. The partitioning of the data into a specified number of mutually exclusive subsets is usually referred as *hard clustering*. In many situations this is not an adequate representation.

Consider for instance a borderline point located in the boundary between two clusters or otherwise an outlier datum located at nearly the same distance from the centers of two clusters. In these frequent situations the point is almost as typical of one cluster as it is of the other, thus a more natural partition would be one which allowed the objects to belong to several clusters simultaneously (with different degrees of membership.) This is precisely the central concept behind *fuzzy clustering* methods with foundations in the fuzzy sets theory [77]. A *fuzzy set* is characterized by a *membership function* that maps each point of the universe $\mathbf{X}$ to a number in the interval $[0, 1]$ (1 represents full degree of inclusion and 0 non-membership at all).

We can say that the relaxation of the constraint imposed on the partition matrix to $u_{ij} \in [0, 1]$ is more realistic and able to provide a richer insight of the data structure, especially when in presence of ambiguous data or clusters without sharp boundaries. Indeed, the fuzzy logic approach to clustering differs from the conventional set theory approach mainly because a generic datum may belong to more than one cluster with a different degree of membership (usually a value between 0, non-membership, and 1, full degree of inclusion). Hence the data points near the core of a given cluster exhibit a higher degree of membership than those lying farther away (near its border). Within this framework it is possible to capture the uncertainty, vagueness and flexibility inherent to the data set and to the concepts being formed. In the remaining of this section we review some different algorithmic approaches that allows the construction of fuzzy partitions, i.e., algorithms which represent a cluster as a fuzzy set.

### 2.1 The Fuzzy C-Means Clustering Algorithm

Fuzzy clustering was introduced as early as 1969 by Ruspini [68]. Fuzzy C-Means (FCM) is a simple and widely used clustering algorithm. The algorithm results from an optimization problem that consists in the minimization, with respect to **V**, the set of prototypes, and **U**, the fuzzy membership matrix, of the following index (objective function) [6]:

$$Q_{FCM} = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i) \tag{3}$$

where $m > 1$ is the so-called fuzziness parameter ($m = 2$ is a common choice) that controls the influence of membership grades or in other words how much clusters may overlap, cf. [44], and $D$ stands for a norm distance in $\mathbb{R}^n$, under the following conditions on the partition matrix elements:

$$u_{ij} \in [0,1] \text{ for all } i = 1, \ldots, C \ \text{ and } j = 1, \ldots, N \tag{4}$$

$$\sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j = 1, \ldots, N \tag{5}$$

$$\sum_{j=1}^{N} u_{ij} > 0 \text{ for all } i = 1, \ldots, C \tag{6}$$

Condition (5) induces a fuzzy partition in the strict sense and assures that every datum has a similar global weight on the data set. Constraint (6) guarantees that none of the $C$ clusters is empty, thus implying a cluster partition with no less than $C$ clusters. Notice the similarity between (2) and (3). As a matter of fact they are coincident apart from a fixed transformation (the introduction of the fuzzifier, $m$) introduced as a mean to prevent that under condition (5) the same minimum as the one obtained by the crisp standard formulation was reproduced.

For this constrained nonlinear optimization problem there is no obvious analytical solution. Therefore the most popular and effective method to minimize the constrained objective function consists in resorting to a technique known as alternating optimization. This means that one set of parameters is kept fixed while the other is being optimized, and next they exchange roles. The prototype **V** and membership **U** update equations are obtained from the necessary conditions of a minimum:

$$\frac{\partial Q_{FCM}}{\partial \mathbf{V}} = 0 \ \ \text{(assuming } \mathbf{U} \text{ to be constant);} \tag{7}$$

$$\frac{\partial Q_{FCM}}{\partial \mathbf{U}} = 0 \ \ \text{(assuming } \mathbf{V} \text{ to be constant).} \tag{8}$$

Additionally the consideration of (5) in the original objective function (3) by means of Lagrange multipliers converts the constrained problem into its constrained-free version. Some straightforward computations lead to the update formula of the partition matrix:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left( \frac{D_{ji}(\mathbf{x}_j, \mathbf{v}_i)}{D_{jk}(\mathbf{x}_j, \mathbf{v}_k)} \right)^{\frac{2}{(m-1)}}}. \tag{9}$$

This formula does not depend on the chosen distance function, however the determination of the prototypes is more complicated since many distance norms do not lead to a closed-type expression. A common practical choice is to use the Euclidean distance or $L_2$ norm (for a generalization to $L_p$, $p > 0$, the interested reader is referred to [38]) leading to the following prototype update equation:

$$\mathrm{v}_i = \frac{\sum_{j=1}^{N} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{N} u_{ij}^m}. \tag{10}$$

The alternate optimization of **U** and **V** proceed iteratively until no significant change of the objective function is registered. It has been proven that the generated sequence of solutions, for fixed $m > 1$ always converge to local minima or saddle points of (3) [9]. Informally, what the resulting algorithm will do is to search for the clusters that minimize the sum of the intra-cluster distances. In general the performance of fuzzy algorithms, when compared with the corresponding hard partitioning ones, is superior and they are less prone to be trapped in local minima [6]. However, like its hard counterpart the FCM algorithm shares the problem of high sensitivity to noise and outliers, something that is common to the generality of the least-squares approaches and that can drastically distort the optimal solution or facilitate the creation of additional local minima. Next we discuss an alternative formulation, specifically designed to tackle this problem.

### 2.2 The Possibilistic C-Means Clustering Algorithm

The influence of noise points can be reduced if the memberships associated with them are small in all clusters. However, as can be seen from the probabilistic-like constraint (5), the memberships generated by the FCM are relative numbers expressing the concept of sharing of each pattern between clusters rather than the concept of typicality of a given pattern to a given cluster. This means that noise points and outliers will also have significantly high membership values. A more general form of fuzzy partition, the *possibilistic* partition, can be obtained by relaxing the constraint (5) in order to address this problem.

In this case the assignment of low membership in each cluster to noise points depends on giving up of the normalization condition (5), leading to possibilistic instead of fuzzy partitions. To avoid the trivial solution (i.e. a matrix with null elements) Krishnapuram and Keller [47] added to (3) a punishment term for low memberships resulting in the augmented possibilistic $c$-means (PCM) objective function:

$$Q_{PCM} = \sum_{i=1}^{C}\sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i) + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1 - u_{ij})^{m} \qquad (11)$$

where the distance parameters $\eta_i > 0$ $(i = 1, \ldots, C)$ are specified by the user. Notice that the second term expresses the desire to have strong assignments of data to clusters. Due to the nature of the membership constraint, we call *possibilistic clustering algorithm* (PCM) a fuzzy clustering algorithm which minimizes (11) under the constraint (6). The partition matrix update equations, as before for the FCM case, are obtained by setting the derivative of the objective function equal to zero while holding the prototype parameters fixed:

$$u_{ij} = \frac{1}{1 + \left(\frac{D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)}{\eta_i}\right)^{\frac{1}{(m-1)}}}. \qquad (12)$$

This update expression clearly emphasizes the typicality interpretation of the membership function. Unlike the FCM formulation, the degree of membership of one point to a cluster depends exclusively of its distance to the center of that cluster. For the same cluster, closer points obtain higher membership than the ones farther away from it. Moreover (12) shows that $\eta_i$ determines the distance of the "definite" assignment ($u_{ij} > 0.5$) of a point to a cluster (simply considering $m = 2$ and substituting $\eta_i$ by $D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)$ results in $u_{ij} = 0.5$). So it is useful to choose each $\eta_i$ separately, according to the individual geometrical features of each cluster. Unfortunately these are not always available so Krishnapuram and Keller recommend several methods to determine $\eta_i$ [47,48]. Using the fuzzy intra cluster distance a sound probabilistic estimation of these weight factors can be obtained:

$$\eta_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)}{\sum_{j=1}^{N} u_{ij}^{m}}. \qquad (13)$$

The update formula for the prototypes is the same as the one used in the FCM method since the second term in (11) simply vanishes when computing the derivative of the objective function with respect to the prototype parameters. If we take a closer look at (12) we see that the membership degree of a pattern to a cluster depends only on the distance of the pattern to that cluster, but not on its distance to other clusters. So it happens that in some situations this algorithm can originate coincident clusters (converging to the same local optimal point), thus disregarding clusters with lower density or less points, or even presents stability problems due to sensitivity to initialization [48]. Thus to overcome these drawbacks of the possibilistic approach it is common practice to initialize PCM with a prior run of the probabilistic FCM.

### 2.3 Other Approaches to Fuzzy Clustering

The literature on fuzzy clustering is remarkably rich, cf. [72], and in a broad sense it reflects the attempts made to surpass the problems and limitations of the FCM and PCM algorithms. In the two former sections we reviewed FCM and PCM and their prototypes's update equations assuming the Euclidean distance as the standard metric. However when combined with a squared error-based objective function this distance induces hyper-spherical clusters. To overcome this geometrical constraint imposed by clustering algorithms based on a fixed distance metric several algorithms using adaptive distance measures have been proposed. Two of the most well known are the Gustafson-Kessel algorithm [32] which replaces the Euclidean distance by the Mahalanobis distance (an interesting generalization of the Euclidean distance) with a specific covariance matrix for each cluster and the unsupervised Gath-Geva algorithm [30] where the distance is based on the fuzzification of the maximum likelihood estimation method. Both of these algorithms are well fitted to find ellipsoidal clusters with varying size and orientation (there are also axis-parallel variants of these algorithms and to some extent they can also be used to detect lines).

In the field of image processing and recognition the geometry of the fuzzy clusters is a key aspect for image analysis tasks. Both FCM and PCM use point prototypes. If we are interested in finding particular cluster shapes, algorithms based on hyper-planar or functional prototypes, or prototypes defined by functions, are a good choice. The distance is no longer defined between two patterns (i.e. a datum and a prototype), instead it is measured between a pattern and a more complex geometric construct. This class of algorithms includes the fuzzy $c$-varieties [7] for the detection of linear manifolds (lines, planes or hyper-planes), fuzzy $c$-elliptotypes [8] for objects located in the interior of ellipses, fuzzy shell clustering for the recognition of object boundaries (e.g. fuzzy $c$-shells [16] in the detection of circles, hyper-quadric shells [45], fuzzy $c$-rectangular shells [40]) and fuzzy regression models [36]. The interested reader may follow a comprehensive explanation of these branch of methods in [41].

In addition to PCM other methods have been proposed in order to improve the robustness of the FCM algorithm to noisy data points and outliers while maintaining the constraint (5) (thus circumventing the problem of cluster coincidence of the PCM approach). For instance the technique presented in [59] and [14] consists in the introduction of an additional *noise cluster* aiming at grouping the points with low probability of belonging to the remaining clusters. This probability depends on the mean value of the squared distances between patterns and the prototypes of the normal clusters. Latter on, this technique was extended in order to accommodate different noise probabilities per cluster [15].

The great majority of the algorithms presented hitherto result from alternating the optimization of the membership functions and prototype locations in an iterative process. Therefore the clustering model constrains (and is constrained to) the particular shapes of the membership functions and the positions of the prototypes to those determined by the updating equations derived from the objective function. However, the user might be interested in the use of a certain type of membership function with more adequate shapes to the problem in question or in certain cluster prototypes satisfying some application-specific needs. The *alternating cluster estimation* (ACE) framework [65] is able to provide, when required, this extra flexibility. In applications such as extraction of fuzzy rules from data, where each fuzzy set should have a clear semantic meaning (for instance associated to linguistic labels like "high" temperature or "about 80" degrees), a convex fuzzy set with limited support may be more preferable than the non-convex membership functions generated by FCM or PCM. Notwithstanding that ACE embodies FCM and PCM as particular instances of the framework, the requirement that the updating equations for the membership function and the prototypes should result from the necessary conditions for local extrema is now rejected and the user is free to choose the pair of updating equations which is better fitted for the problem at hand. At first sight this generalization may seem to be lacking mathematical soundness however it has proven its usefulness in practical examples.

In many practical applications the data sets can be heavily contaminated by noise points which promote the proliferation of local minima. In these cases, the probability of the alternate optimization getting stuck at local optimal values is far from being negligible. To obviate this problem, stochastic algorithms have been used in cluster analysis, many of them inspired on biological paradigms such as the natural evolution of species or swarm-based behavior. Examples of such approaches to fuzzy clustering include the use of genetic algorithms [18,19,35,43,51,73], evolutionary programming [69], evolutionary strategies [76], ant colony optimization [66] and particle swarm optimization [67]. Notwithstanding that these attempts do not guarantee optimal solutions, demand the definition of a set of problem-specific parameters (e.g. population size) and are very computationally time-consuming they can undoubtedly contribute to avoid local extrema and reduce the sensitivity to initialization.

### 2.4 Determination of the Number of Fuzzy Partitions

In the great generality of the partitional algorithms the number of clusters $C$ is the parameter having greater influence on the resulting partition. The chosen clustering algorithm searches for $C$ clusters, regardless of whether they are really present in the data or not. So when there is no prior knowledge about the structure of the data a natural question arises: what is the right number of clusters for a particular data set? This question is known in the literature as the cluster validity problem and distinct validity measures have been proposed in order to find an answer, cf. [27, 34, 42, 55, 58, 75]. However, in spite of a greater practical adhesion to some of them, due to the subjective and application-dependent character of the problem there is no consensus on their capability to provide a definitive answer to the foregoing question. For partitional fuzzy clustering it is advisable that the validity indices account both for the data set (e.g. their variance) and the resulting membership degrees. An example of such class of validity indices, exhibiting good behavior when matched against a set of other indices [60], is the Xie-Beni index [74], also known as the compactness and separation index, computed as the ratio of the compactness of the fuzzy partition of a data set to its separation:

$$XB = \frac{\sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} D_{ji}^{2}(\mathbf{x}_j, \mathbf{v}_i)}{N \min_{i \neq j} D_{ij}^{2}(\mathbf{v}_i, \mathbf{v}_j)}. \tag{14}$$

The interested reader is referred to [33] for further examples and properties of hard/fuzzy validation indices. The effectiveness of a particular choice of $C$ is verified *a posteriori* by *cluster validity analysis*, performed by running the clustering algorithm for different values of $C$, several times with different initializations. However, since different validity measures may produce conflicting results (even runs with different initializations may introduce some distortion for the same measure) it is advisable that they should be used only as guidelines to find a plausible range for the correct number of clusters.

The cluster validity problem was also tackled by unsupervised techniques with no *a priori* assumption on the number of clusters. Many of these approaches (e.g. [28,29,53]) take advantage of the fact that (3) is minimized when the number of clusters is equal to the cardinality of the data set (when prototypes and data coincide) by adding to the cost function (3) a regularization term which is minimized when all the patterns are assigned to one cluster. These algorithms start with a large number of clusters which is progressively reduced until convergence. Regretfully, in practice the problem of cluster validity is replaced by the determination in advance of another user supplied parameter with major influence in the clustering outcome and dictating which clusters are discarded.

An interesting blending between fuzzy partitional clustering techniques and hierarchical algorithms was presented in [31]. The objective is to exploit the advantages of hierarchical clustering while overcoming its disadvantages in dealing with overlap between clusters. At every new recursive agglomerative step the proposed algorithm adaptively determines the number of clusters in each bifurcation by means of a weighted version of the unsupervised optimal fuzzy clustering algorithm [30]. The final outcome of the clustering is the fuzzy partition with the best validity index value. Needless to say, the algorithm presents sensitivity to the adopted validity index.

Unsupervised stochastic techniques have also been applied to cluster validity analysis. In [56] a genetic fuzzy clustering algorithm is used for the classification of satellite images into different land cover regions. The objective function is replaced directly by a validity index (in this case the Xie-Beni index) and a variable chromosome length (depending on the number of clusters represented by each individual) allows the simultaneous evolution of solutions with a different number of clusters. The outcome is the best (in the Xie-Beni sense) of the evaluated fuzzy partitions.

## 3 The Role of Fuzzy Clustering in the Human-Centric Paradigm

The concept of *linguistic variable* [79, 80] plays a pivotal role in the formation of fuzzy information granules. Informally, a linguistic variable is a granulated variable whose granular values are words or phrases represented by fuzzy sets (altogether with their connectives, modifiers and negation). These linguistic characterizations are, usually, less specific than the numeric ones, but in compensation are safer. Thus the linguistic variable can be viewed as a way to accomplish (lossy) compression of information. Moreover the linguistic variable provides a descriptive mean for complex or poorly understood systems and, more important, offers a bridge between linguistics and computation, cf. [81]. As Zadeh [83] pointed out, *the fuzziness of granules, their attributes and their values is a central characteristic of the ways in which human concepts are formed, organized and manipulated.* This observation supports what seems to be one of the most human-centric approaches to discover structure in data: fuzzy clustering.

As previously referred, the contribution of fuzzy clustering techniques to the human-centric paradigm effort can be described across three main lines: (i) user-friendly communication of the results, (ii) seamless integration of human knowledge and (iii) incorporation of human-defined semantic interpretability constraints in order to accomplish a transparent system with good accuracy. The purpose of this section is to present a detailed discussion on the two first aspects. The incorporation of human-defined semantic constraints into the clustering endeavor is addressed in Sect. 4.

### 3.1 Information Granulation

Information granules are simultaneously a mean and an objective. Due to the limited capability of human mind and sensory organs to deal with complex information its decomposition into manageable chunks of information is essential. The aggregation of similar or nearby objects into information granules (class abstraction) as well as the encapsulation of functional commonalities are fundamental skills for a successful approach to the great majority of problems that we face everyday. This granulation may be crisp or fuzzy.

Crisp granules are derived with the apparatus of the classical set theory and are common components in various methods of information analysis, e.g. decision trees, interval analysis or rough set theory. Fuzzy granules found their inspiration in the human capability to reason in an uncertain and imprecise environment and are supported by the theory of fuzzy information granulation (TFIG) [82], a part of the fuzzy sets and fuzzy logic framework. Furthermore the fuzzy logic approach relies on the notion of (fuzzy) set, opposite to the member of a classical set, to represent uncertain and imprecise knowledge. This last facet is the point of departure to the model identification with different levels of descriptive precision and granularity, viz. (fuzzy) granulation, cf. [78, 81]. In this setting, typically, an information granule is a fuzzy set and the process of information granulation

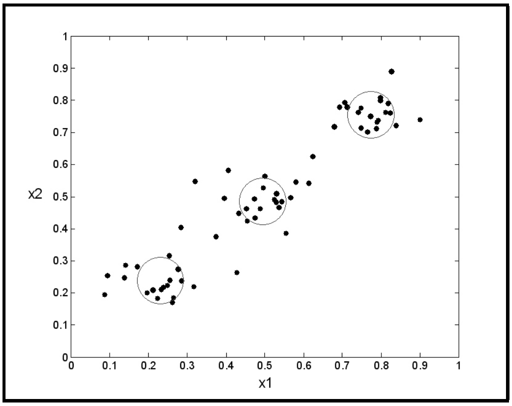


**Fig. 1.** Simple data set in $\mathbb{R}^2$ and clustering results of the FCM algorithm. Dots represent data points and unfilled circles represent the clusters' centers.

consists in describing a crisp or fuzzy object as a collection of fuzzy granules (or eventually as relationships between them).

In Sect. 2 we reviewed the standard fuzzy c-means algorithm (FCM), its assets and common alternatives to overcome its shortcomings. Next with the help of a visually appealing example the path leading from raw data to information granules is briefly explained. To facilitate the visualization we consider a synthetic data set defined in $\mathbb{R}^2$ as depicted in Fig. 1.

It is composed of three visually separable clusters resulting from a normal distribution of twenty elements around 3 distinct points. Suppose that the clusters' centers, marked as unfilled circles in Fig. 1, were found by an adequate fuzzy clustering method (in this case FCM). The purpose here is to describe those clusters invoking simple fuzzy granules. Let's assume that the clustering algorithm has produced the partition matrix where each data point is characterized by a set of membership values, one per each cluster: the closer the point is to the cluster's center, the higher the membership value of that point. This relation can be perceived in Fig. 2 where only the maximum value of membership for each data point is shown (in the $Z$ axis).

Each one of the resulting clusters may be conceived as a multidimensional granule, however to be clearly understandable and subject to human communication it has to be expressed in terms of simpler qualitative attributes defined for each feature.

To accomplish this, first the dimensionality of this fuzzy relation is reduced by a simple operation of projection to the corresponding coordinate spaces. For every 2-dimensional granule $G$ defined on $\mathbf{X_1} \times \mathbf{X_2}$ there are two projections $G_{proj\mathbf{X_1}}$ and $G_{proj\mathbf{X_2}}$ with the following membership functions (for discrete sets $sup$ is replaced by $max$):

$$G_{proj\mathbf{X_1}}(a) = \sup_{y \in \mathbf{X_2}} G(a, y), \quad a \in \mathbf{X_1}; \tag{15}$$

$$G_{proj\mathbf{X_2}}(b) = \sup_{x \in \mathbf{X_1}} G(x,b), \quad b \in \mathbf{X_2}. \tag{16}$$

Computing the correspondent projections, each cluster induces a one-dimensional *discrete* fuzzy set per feature. Figure 3 depicts the projection into one of the coordinate spaces (notice that for ease of visualization the individual fuzzy sets are depicted as piecewise linear functions when, in fact, they are composed of discrete elements).

To extend this fuzzy set to the whole one-dimensional domain an adequate enveloping fuzzy set (convex completion) or a suitable parameterized fuzzy set approximation is usually necessary. Obviously this approximation implies some

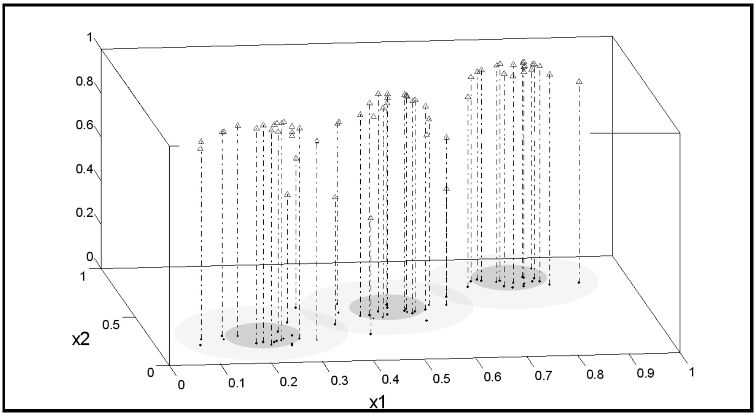


**Fig. 2.** Clustering results of the FCM algorithm depicting the maximum membership value of each data point

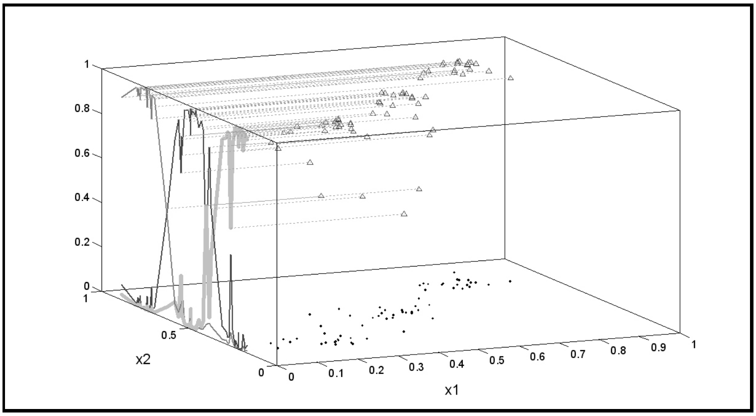


**Fig. 3.** Dimensionality reduction by projection to the coordinate spaces. The figure depicts the projection to $\mathbf{X_2}$.

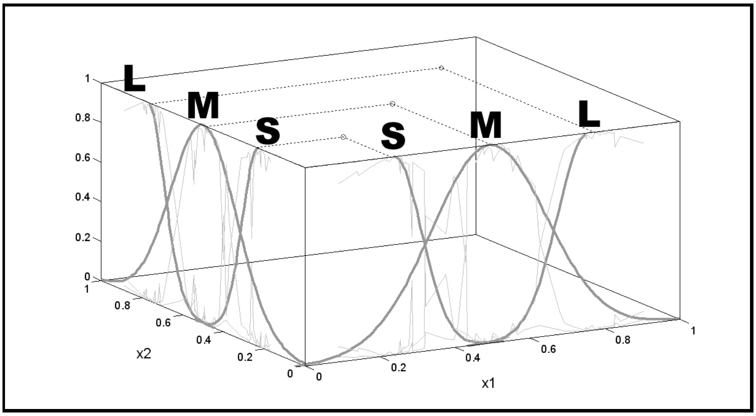


**Fig. 4.** Synthesis of interpretable Information Granules. A functional approximation was performed followed by a semantic conversion conveying meaning to each one-dimensional fuzzy set.

loss of information. In the given example each one-dimensional fuzzy set was approximated by a Gaussian membership function distributed around the prototype's projection, see Fig. 4, and altogether they form a fuzzy partition across each single domain. Finally a last step must be performed if one wishes to describe each multidimensional granule in an human-friendly way: if possible each one-dimensional fuzzy set must be associated to a linguistic value with a clear semantic meaning (in Fig. 4, $S$ stands for Small, $M$ for Medium and $L$ stands for Large).

The multidimensional granule is thus defined as a combination of one-dimensional fuzzy sets encoding linguistic labels relevant for the problem at hand. For each cluster the one-dimensional fuzzy sets where its prototype's projection attains the maximum value are chosen as the clusters' representatives. Hence each multidimensional cluster may be expressed as cartesian products of simpler granules. Referring back to Fig. 4 the overall data set may be entirely described in this concise form:

$$S_1 \times S_2 + M_1 \times M_2 + L_1 \times L_2 \tag{17}$$

where + represents disjunction and $X_1$ and $X_2$ play the role of linguistic variables assuming the values small ($S$), medium ($M$) and large ($L$), necessarily with different concretization in each domain.

### 3.2 Elicitation of the Information Granules

Fuzzy clusters are information granules represented by fuzzy sets, or more generally by fuzzy relations in some multi-dimensional data space. However, as was emphasized above, in order to take full advantage of their expressive power they should be able to be described as propositions in a natural language.

In opposition to our previous oversimplified example (Figs. 1, 2, 3 and 4) there are many situations posing several difficulties to the adequate elicitation of a *semantic mapping* between data space and feature space. Just to give an example, consider a data set with 4 well separable clusters in $\mathbb{R}^2$ and centers in the vicinity of the vertices of a square with sides parallel to the coordinate axes. In this case the correspondent projections into the one-dimensional spaces, would result in two pairs of very close fuzzy sets per feature and consequently almost undiscernible between them. The approaches to develop semantically sound information granules as a result of the fuzzy clustering process range from purely *prescriptive* methods to purely *descriptive* techniques, cf. [63]. In the prescriptive characterization of the fuzzy sets the meaningful granules are expressed intuitively by an observer in such a way that they capture the essence of the problem. The descriptive design involves the detailed computation of the membership functions based on the available numeric data.

The work presented in [12] is a good example of this latter approach complemented with some semantic concerns. The overall technique can be summarized in three steps. First, a cluster analysis is performed on the data set. The clustering algorithm (e.g. FCM) induces $C$ information granules and this number of clusters has a major effect on the information granularity. In the second step the prototypes are projected into each dimension, being their projections further clustered in order to obtain a pre-specified number of clusters, i.e., one-dimensional granules. The final step consists in quantifying the resulting one-dimensional prototypes as fuzzy sets in the feature space by means of Gaussian membership functions with a desired level of overlap. The second step of this double-clustering technique is not computationally demanding (the number of prototypes is much lesser than the number of data elements) and promotes the fusion of projections, which otherwise would result in undiscernible data sets, into one single representative thus permitting the representation of granules via highly comprehensible fuzzy propositions.

The prescriptive approach can be illustrated by the interesting technique of context clustering [61] (see also [39]). In essence the algorithm results from an extension to the FCM algorithm replacing the standard normalization constraint (5) by a conditioning constraint dictated by the context under which the clustering is being performed. The context is specified by the user and can assume the form of an information granule (linguistic term) defined in a peculiar feature or a logical combination between granules in the same feature space or even a composite context resulting from the Cartesian product of fuzzy sets defined in different feature spaces. Informally, we can say that the (fuzzy) linguistic context acts as a data window focusing the clustering effort on particular subsets of the data or regions of interest, thus enabling a deeper insight on the internal structure of those information granules.

The technique reported in [63] tries to present a balanced tradeoff between the prescriptive and descriptive approaches. The descriptive component is represented by the clustering algorithm (the experiments report to the standard FCM) performed in the multi-dimensional data space. Given two different runs of the clustering algorithm, searching for a different number of clusters, the resulting granules present necessarily a different granularity level. The distinct granularity of the resulting information granules (the mixture of coarser and finer granules) can be turned into an advantage. The prescriptive component task is to conciliate different granular representations by means of specific operations of generalization (information granules combined *or*-wise) and specialization (information granules refined *and*-wise) of the fuzzy relations. The logic operators (s-norms and t-norms) are defined in advance then, if we intend to decrease the granularity of the finer result, the algorithm finds the coarser granule and the respective generalization (selected amongst the possible pairwise generalizations of the finer granular set) with optimal similarity (based on the overall difference between membership values of each datum in the original set and in the given generalization). On the other hand, when we intend to increase the granularity of the collection of information granules a similar process is performed, viz., the optimal replacing of a granule by the pair of granules forming its closest specialization.

The representation of information granules via multidimensional hyper-boxes with sides parallel to the coordinates greatly simplifies their transparent expression as decomposable relations of classical sets in the corresponding feature spaces. In [3] the standard FCM was modified through a gradient-based technique in order to accommodate the Tchebyshev distance. This distance induces a hyper-box shaped geometry of the clusters, however due to the interaction between clusters there exists a deformation of the hyper-boxes which need to be reconstructed in an approximate manner. When compared with FCM with Euclidean distance, the algorithm equipped with Tchebyshev distance exhibited less sensitivity to the size of the data groupings, being able to identify smaller clusters. The algorithm produces a description of the data consisting of hyper-boxes (whose sizes depend on a given threshold) which encompass the core of the data and a residual portion of the data described by the standard FCM membership expression. Another interesting approach to hyper-box granulation combined with fuzzy clustering was presented in [2]. The proposed measure of information density (the ratio between cardinality and specificity of a set) is maximized in a recursive manner departing from the numeric data which is progressively mixed with the produced granular data. As a result of this granulation process the data is compressed while the number of information granules in the high data density areas is reduced. Next the information granules are clustered using the FCM algorithm combined with a parametric method of representation of the hyper-boxes. This results in a collection of cluster prototypes interpretable in the original data space as hyper-boxes altogether with a fuzzy partition matrix representing the membership of data into clusters. Due to the reduction of the number of information granules in high density areas the FCM problem of under-representing smaller groups of data is thus obviated. Moreover, the hyper-box representation of the prototypes has direct transposition as fuzzy decomposable relations in the feature space enabling a transparent interpretation of the information granules.

Independently of the followed approach (descriptive, prescriptive or both) one should be aware that the elicitation of information granules in a human comprehensible way is dependent of the characteristics of the application at hand and on the judicious decisions of the data analyst.

### 3.3 Enhancements to Accommodate Human Knowledge

Notwithstanding that in a great number of applications there is no labeling information about the data at hand, or otherwise the labeling of the data set is a fastidious task requiring a lot of expert time due to its high cardinality, or even it is error prone and potentially ambiguous due to the nature of the problem being tackled; there are situations where the clustering endeavor can and should be guided, at least in a confined way, by the inclusion of additional information about the data structure and the inherent characteristics of the problem. Automatic text classification of extensive corpora, categorization of Web sources, recognition of handwritten text characters or image segmentation are just some examples of applications where usually the data analyst is confronted with a small subset of labeled data. Better than blindly attack the problem as an unsupervised clustering pursuit the available information should be properly taken into account.

In cases like the ones mentioned we can say that we are faced with a semi-supervised clustering and as was emphasized in [5] the labeled data can be used quite successfully not only to define the number of clusters but also by using the cluster centers as a way to affect the cluster assignment for unlabeled data. A very straightforward formulation that allows us to play with the balance between our confidence in the reliability of the available labeled data and the automated data exploration was presented in [64]. Simply stated the partially supervised clustering algorithm lays on an additive objective function aiming at structure finding, minimizing the fuzzy within cluster variance as in the standard FCM, and accounting for the data already categorized by minimizing the misclassification error:

$$Q_{semi} = Q_{FCM} + \alpha \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij}^{l*} - u_{ij}^{l})^m D_{ji}^2(\mathbf{x}_j, \mathbf{v}_i) \tag{18}$$

Here $l$ alludes to the subset of classified patterns and $u_{ij}^{l*}$ stands for the partition matrix containing membership grades previously assigned to the selected patterns, possibly by a domain expert. The optimization of $u_{ij}^{l}$ intends to make it close to the information already available. The nonnegative regularization factor $\alpha$ defines the balance between the supervised and unsupervised learning. Higher the value of $\alpha$, higher our beliefs in the labeling decisions already made and consequently our willingness to overlook some level of structural optimization. As previously mentioned, being an optimization scheme that relies on alternate optimization, it can be trapped by local optima and is also very sensitive to initialization. Latter on this problem was ameliorated for this specific case, by using evolutionary optimization techniques [50] obviously at the expense of a great deal of computational cost.

When one does not have the knowledge about how many data classes are there but is still able to provide some indications on the resemblance or proximity of some data points, this information can be captured for instance with the help of another interesting approach presented in [52]. The underlying principle of the proximity fuzzy C-means (P-FCM) is the guided collaboration between data processing and knowledge processing through the consideration of proximity constraints. These are expressed as a symmetric and reflexive mapping of pairs of patterns to the unit interval (its value is maximal when two patterns are coincident). The relation between the fuzzy partition matrix produced by the clustering algorithm and the proximity mapping is set up by the following expression:

$$\hat{p}[k_1, k_2] = \sum_{i=1}^{C} min(u_{ik_1}, u_{ik_2}) \tag{19}$$

This expression is used to build the deduced symmetric proximity matrix. The algorithm consists of two main phases that are realized in interleaved manner. The first phase has a data driven nature and consists in applying the standard FCM to the patterns. The second concerns the accommodation of the proximity-based hints and involves some gradient oriented learning. In order to guide the gradient search procedure the objective function for the second phase penalizes the differences between the available proximity levels and the corresponding ones from the deduced proximity matrix. Since in fuzzy relational clustering, cf. [37,46], the data is described by specifying pairwise similarities or dissimilarities between objects, at first sight it seems that there is some resemblance between this algorithm and relational clustering pursuits. However it is worthwhile to note that in this case one is provided with object data describing the objects through feature measurements and using this technique, or for that matter other with similar inspiration, it is possible to conciliate strictly structural algorithmic information with the available relational information (if there is any).

## 4 Deploying Semantic Constraints for Data Clustering

The need for semantically valid fuzzy systems (classifiers, models, controllers) is a matter of common agreement nowadays. A set of semantic properties, necessary to ensure an interpretable fuzzy system during system design, have been outlined and analyzed [70, 62, 71]. Based on these properties, several different constraints have been derived and applied to several real-world situations, including [21,22,24,25]. For a more complete and recent survey of the literature on these matters the reader is referred to [10,11]. The great majority of the reported work on semantic concerns can be classified as belonging to the supervised learning or optimization schemes. Typically, the semantic conditions become part of the learning rule of a neural network, or part of the fitness function of a clustering algorithm, or becomes an optimization goal of its own in multi-objective optimization schemes. In this section we elaborate on a set of semantic constraints easily justifiable at the pure human-centric conceptual level, describing them in a more sound formal framework and demonstrating that they generalize a set of constraints commonly employed in partitional fuzzy clustering. Later the evolutionary semantic driven (ESD) fuzzy clustering algorithm [23] is contextualized as an illustrative example of employing such human-defined constraints where those are used as the defining characteristic which enables the correct identification of the number of clusters present in the data set.

### 4.1 Translation of Semantic Constraints to Clustering

The ultimate objective of clustering is the description of the inherent structure of the data in a comprehensible way. Fuzzy clustering algorithms apport a valuable surplus when aiming at such goal since the identification of regions of interest of a data set can be transposed to propositions on meaningful linguistic labels, thus facilitating the empirical semantic validation of the model.

This translation is highly dependent of the semantic soundness of the fuzzy sets in the distinct feature spaces. In this respect the set of semantic properties postulated in [71], in the context of fuzzy modeling, can be adopted as useful guidelines. These properties emerged as a mean to clarify the meaning of a linguistic term (a fuzzy set) when matched against other linguistic terms in the same universe of discourse. The proposed set of properties includes: a moderate number of membership functions, coverage, normality, natural zero positioning and distinguishability.

Three of these properties seem to have an inherent interest for the clustering endeavor:

1. A moderate number of membership functions since although this number is clearly application dependent, if one intends to describe the structure of the data in a human-friendly way there are strong reasons for imposing an upper bound on the number of clusters (in the limit, when the number of membership functions approaches the cardinality of the data, a fuzzy system becomes a numeric system). This constraint makes sense not only in the feature space, where the typical number of items efficiently handled at the short-term memory (7±2) [57] can be adopted as the upper limit of linguistic terms, but also in the data space since a high number of clusters result in information granules with a high granularity level.
2. Coverage which states that membership functions should cover the entire universe of discourse, so that every datum may have a linguistic representation.
3. Distinguishability since this property is clearly related with cluster separation (membership functions should be distinct enough from each other).

It is worth noting that these properties can be ensured using a variety of constrains, hereafter referred as interpretability constraints.

Consider the sigma-count operator, $M_p(\mathcal{L}_x)$, defined as follows:

$$M_p(\mathcal{L}_x) = \sqrt[p]{u_1^p + \ldots + u_C^p} \tag{20}$$

where $\mathcal{L}_x$ is a fuzzy set representing a real-valued pattern $x$ from the data set, $u_i$ $(i = 1, \ldots, C)$ is the membership degree of $x$ in the $i$-th cluster, $p$ being a positive integer.

Therefore, in the clustering context coverage can be formalized in the following way:

$$\forall_{x \in \mathbf{X}} M_p(\mathcal{L}_x(x)) > 0 \tag{21}$$

Alternatively, one can ensure coverage using the concept of optimal interfaces [70]. Let $\mathbf{X} = [a, b] \in \mathbb{R}^m$. The ordered pair $(\mathcal{L}_x, \mathcal{N}_x)$ is said a pair of optimal interfaces *iff*

$$\forall_{x \in \mathbf{X}} \mathcal{N}_x(\mathcal{L}_x(x)) = x \tag{22}$$

where $\mathcal{L}_x : \mathbf{X} \rightarrow [0, 1]^n$ is the mapping provided by the input interface of the variable $x$ and $\mathcal{N}_x : [0, 1]^n \rightarrow \mathbf{X}$ with $\mathcal{N}_x([0, \ldots, 0]) = \emptyset$ is the mapping provided by the output interface associated with the variable $x$.

Analogously distinguishability can also be enunciated with the help of the sigma-count measure:

$$\forall_{x \in \mathbf{X}} M_p(\mathcal{L}_x(x)) \leq 1 \tag{23}$$

The rationale for this constraint is straightforward: if we have two clusters "very close" to each other eventually there will be points in between the clusters' centers with high membership in both clusters. If the clusters are far apart, then there should not be such cases. Next we show that these constraints generalize the common constraint (5).

**Proposition 1.** *A partition matrix* $\mathbf{U}$ *represented by a class of membership degrees satisfying the constraint (23) and the optimal interface definition*

$$\forall_{j \in \{1, \ldots, N\}} \mathcal{N}(\mathcal{L}(x_j)) = x_j \tag{24}$$

*generalizes the constraint (5).*

*Proof.* Observe that $\mathcal{L}(x_j) \triangleq [u_{1j}\ u_{2j}\ \ldots\ u_{Cj}]'$. Consider by absurdity that $\exists_{s \in \{1, \ldots, N\}} \forall_{i \in \{1, \ldots, C\}} : u_{is} = 0$, or equivalently $\exists_{s \in \{1, \ldots, N\}} : \mathcal{L}(x_s) = \emptyset$. However, by definition, the output mapping $\mathcal{N}$ is undefined for the empty set, thus $\mathcal{N}(\mathcal{L}(x_s)) \neq x_s$, which contradicts (24).

Therefore $\forall_{j \in \{1, \ldots, N\}} \exists_{i \in \{1, \ldots, C\}} : u_{ij} > 0$, implying that

$$\sum_{i=1}^{C} u_{ij} > 0 \text{ for all } j = 1, \ldots, N. \tag{25}$$

If $\mathbf{U}$ satisfies the distinguishability constraint then

$$\forall_{j \in \{1, \ldots, N\}} M_p(\mathbf{u}_j) \leq 1 \tag{26}$$

where $\mathbf{u}_j \triangleq [u_{1j}\ u_{2j}\ \ldots\ u_{Cj}]'$, or in an equivalent way:

$$M_p(\mathbf{u}_j) = \left( \sum_{i=1}^{C} u_{ij}^p \right)^{1/p} \leq 1,\ j = 1, \ldots, N \tag{27}$$

Obviously for $p = 1$, $M_1(\mathbf{u}_j) = \sum_{i=1}^{C} u_{ij} \leq 1$, that together with (25), i.e., $0 < \sum_{i=1}^{C} u_{ij} \leq 1$ generalizes the common constraint (5). □

Notice that $p = 1$ determines the strongest constraint whereas $p = \infty$ describes a loose constraint. Actually it is straightforward to verify that [70]

$$M_1(\mathbf{u}_j) \geq M_2(\mathbf{u}_j) \geq \ldots M_\infty(\mathbf{u}_j) = H(\mathbf{u}_j) \tag{28}$$

where $H(\mathbf{u}_j)$ denotes the height of the fuzzy set $\mathbf{u}_j$, i.e., its maximum membership degree.

With this formulation it is clear that:

1. $\forall_{j\in\{1,\ldots,N\}}$ there is always some non-null membership degree in some cluster.
2. Given a cluster $i$, $i \in \{1, \ldots, C\}$, there is no guarantee that it has elements, so it may happen that $\sum_{j=1}^{N} u_{ij} = 0$.

In a first analysis, this can be obviated if a penalty term on the number of clusters is included on the cost functional or if we allow that the optimization process may also determine the number of clusters. In this work we followed the second approach as described in the following sections.

## 4.2 Evolutive Semantic Driven Fuzzy Clustering

Evolutionary Algorithms (EAs) are adaptive robust methods widely applicable to search, optimization and learning problems [13, 17]. EAs require a limited amount of knowledge about the problem being solved. Relative evaluation of the candidate solutions is enough and no derivatives of cost functions are required. The evolution of the potential solutions over successive generations comprises different phases. Generally speaking, the first phase involves the quantitative evaluation of each individual in the population. This value determines the probability that an individual has to be selected and to carry its genetic material for the next phase. In the second phase, the selected individuals (potential solutions) are given the chance to mate and exchange genetic material with other individuals by means of a crossover operator. The result of this reproduction phase is a new offspring population, which replaces (or sometimes compete with) the previous population. Some of these newly born individuals were possibly prone to some mutations. This process continues until a stop criterion has been met.

### Chromosome Representation

Evolutionary algorithms, being a general optimization strategy, can be adapted to objective function based fuzzy clustering. Obviously, it is necessary to find an adequate representation for the parameters to be optimized, viz. the prototypes and the membership degrees of the partition matrix. However the simultaneous optimization of the $C$ prototypes and the $C \times N$ membership degrees seems, to say the least, unpractical. Thus we restricted the complexity of the optimization task by optimizing only the prototypes and computing the corresponding membership degrees using the updating expression of the PCM model (12) which, as was previously said, does not impose a strict fuzzy partition since the degree of membership of each point depends exclusively of its distance towards the prototype.

### Fitness Assignment

In the presented evolutionary semantic driven fuzzy clustering algorithm the quantitative assessment of each potential solution is based on the sum-of-squares criterion (3) but introduces two extra terms in order to ensure the above mentioned properties of coverage and distinguishability:

$$\begin{aligned} Q_{cov} &= \textstyle\sum_x \|x - x^*\|^2 \\ Q_{dist} &= \textstyle\sum_x [(M_p(\mathbf{u}_j) - 1)^2 step(M_p(\mathbf{u}_j) - 1)] \end{aligned} \tag{29}$$

where *step* represents the unit step function (equals 1 if its argument is greater than zero and equals 0 otherwise). The index, $Q_{cov}$, based on the concept of optimal interfaces [71], seeks an adequate coverage level. To compute $x^* = \mathcal{N}_x(\mathcal{L}_x(x))$ the internal representation $\mathcal{L}_x$ was specified in agreement with (12) and the output interface $\mathcal{N}_x$ was given by the center of gravity defuzzification method:

$$x_j^* = \frac{\sum_{i=1}^{C} u_{ij}\mathbf{v}_i}{\sum_{i=1}^{C} u_{ij}} \tag{30}$$

The index $Q_{dist}$ is intended to keep the clusters prototypes apart from each other. This implies that the points with a sigma-count above the unity become penalized.

A very interesting side effect of these indices, is that in a variable number of clusters optimization environment $Q_{dist}$ attains its minimum for the single cluster case, $C = 1$, whereas $Q_{cov}$ attains its minimum when the number of clusters is equal to the number of patterns, $C = N$. To fully comprehend the importance of this side effect, one should recall that in a variable number of clusters optimization scheme, $Q_{FCM}$ (3) also attains its minimum at $C = N$. Therefore, when both three criteria are considered, a balance is obtained for some $C$ in between 1 and $N$. In the reported experiments we will demonstrate that this tradeoff can be used in our advantage as a mean of determination of the number of clusters.

The quantitative assessment of each individual in the population is given by the following cost functional:

$$Q_{ESD} = Q_{FCM} + Q_{cov} + Q_{dist} \tag{31}$$

Thus the variable length chromosome, exhibiting a set of real coded cluster prototypes which result in low intra-cluster distance, combined with an adequate coverage level (and accurate positioning) altogether with clearly distinct clusters obtains an higher fitness value than other lacking any of these three desired features.

Since EAs are designed for maximization problems, the current minimization problem was converted into a maximization one using the following transformation:

$$fitness(x) = \frac{1}{1 + Q_{ESD}(x)} \tag{32}$$

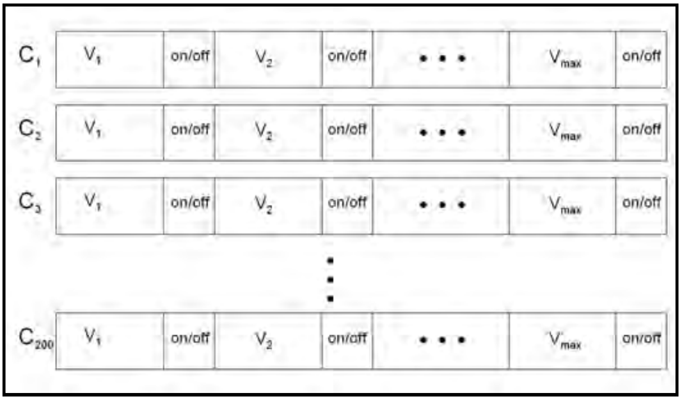
We would like to stress that in the expression (31) a set of variable weights could be considered in order to give different importance to each of the three terms, however such study is beyond the scope of this work. In the reported experiments we treated each objective in the same way: for each chromosome its value was calculated, then the whole range of values across the entire population of chromosomes is translated to the unit interval and only after this normalization the three objectives were summed up as in (31).

### 4.3 Numerical Examples

A collection of synthetic data sets presenting distinct challenges for the clustering process was used to demonstrate the viability of our approach. In order to simplify the visualization in these examples the patterns are distributed in $\mathbb{R}^2$. Moreover we present the results obtained from two *de facto* benchmark data sets from the clustering literature: the Ruspini data set [68] and the Iris Plants data set [1]. The presented results refer to mean values of ten independent runs per data set.

The parameters of the EA were kept constant in the presented experiments. A stopping criterion of 1000 iterations was used. The population was composed of 200 chromosomes, each one representing a set of real-coded cluster prototypes, as depicted in Fig. 5.

The maximum number of prototypes per chromosome was dependent on the number of patterns and fixed on the common heuristic approximation for the maximum number of clusters in a data set, viz. $\sqrt{N}$. Each cluster prototype had



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $C_1$ | $V_1$ | on/off | $V_2$ | on/off | • • • | $V_{max}$ | on/off |
| $C_2$ | $V_1$ | on/off | $V_2$ | on/off | • • • | $V_{max}$ | on/off |
| $C_3$ | $V_1$ | on/off | $V_2$ | on/off | • • • | $V_{max}$ | on/off |
| ⋮ | | | | | | | |
| $C_{200}$ | $V_1$ | on/off | $V_2$ | on/off | • • • | $V_{max}$ | on/off |

**Fig. 5.** Graphical representation of the population

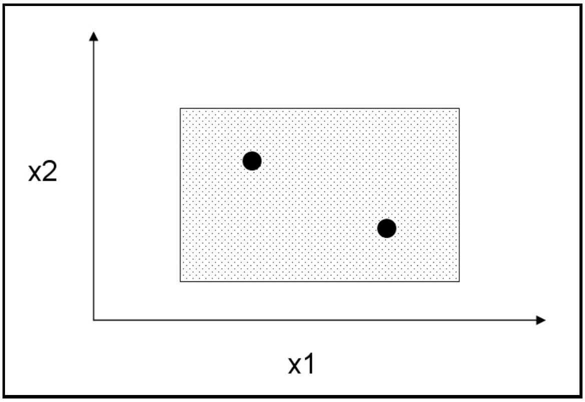


**Fig. 6.** The BLX-$\alpha$ crossover operator. The big dots represent the parents, while the small dots indicate possible children.

a bit field indicating whether or not it was active, allowing a variable number of clusters representation. During crossover this bit field was exchanged between parents – as we will see the crossover of the prototypes was more elaborated.

The selection operator applied in the experiments was stochastic sampling with replacement also known as the roulette wheel selection method. This sampling method selects parents according to a spin of a weighted roulette wheel. The high-fit individuals will have more area assigned to them on the wheel and hence, a higher probability of being selected to the mating pool where they are combined with other individuals by means of a crossover operator.

The blend crossover operator, BLX-$\alpha$, is specifically designed for real-valued chromosomes [20] and was used to combine the genes encoding the prototypes. The resulting offsprings are distributed across a hyper-box encompassing the two parents. The parameter $\alpha$ extends the bounds of the hyper-box, hence to the children is given the possibility to explore new search space inside of an extended range given by their parents, see Fig. 6.

Each real-valued component, $x_i$, of the new offspring is randomly chosen (with an uniform distribution) from the interval $[x_{imin} - I.\alpha,\ x_{imax} + I.\alpha]$, where $x_{imin} = min(x_i^A, x_i^B)$, $x_{imax} = max(x_i^A, x_i^B)$ and $I = x_{imax} - x_{imin}$ with $A$ and $B$ denoting the two progenitors. In fact BLX-$\alpha$ is designed to promote diversity, greater with the increase of $\alpha$, counteracting the decrease in variance that results from the application of the selection operator hence preventing premature convergence of the population. In this work the BLX-$\alpha$ crossover operator was applied with 0.9 probability and the parameter $\alpha$ was set to 0.5 – a common choice in the literature.

The probability of the uniform mutation operator was set to 0.05. In order to prevent that good solutions disappear from the population an elitist approach was used. For each $c \in \{2, \ldots, \sqrt{N}\}$ the best chromosome with $c$ active prototypes was preserved for the next generation. The remaining parameters of the

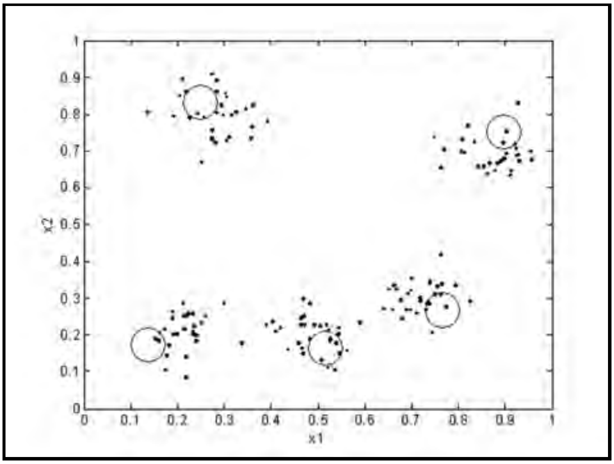


**Fig. 7.** Data set with 5 clusters each one comprising 30 patterns. The unfilled circles mark the prototypes.

algorithm were defined as follows: $p = 1$ in (29); $m = 2$ in (3), (12) and (13). In the presented experiments we used the Euclidean distance as the measure of dissimilarity.

## Example 1

In this example the data set is composed of five visually separable clusters resulting from a normal distribution of thirty elements around 5 distinct points (Table 1 presents the details of the distribution).

Figure 7 presents the data set and the means of the cluster prototypes (unfilled circles) for ten distinct runs.

It is interesting to refer that the algorithm was able to find the number of focal elements used to generate the data set in every run. Also the positioning of the centers of the clusters reveals a noticeable feature of the algorithm: the distinguishability index is contributing to maintain the areas of influence of the clusters apart from each other.

**Table 1.** Details of the normal distribution of the data set of Example 1 (5 clusters)

| Cluster | #Points | Mean | Standard Deviation |
|---|---|---|---|
| 1 | 30 | (0.3; 0.8) | (0.05; 0.05) |
| 2 | 30 | (0.2; 0.2) | (0.05; 0.05) |
| 3 | 30 | (0.5; 0.2) | (0.05; 0.05) |
| 4 | 30 | (0.7; 0.3) | (0.05; 0.05) |
| 5 | 30 | (0.9; 0.7) | (0.05; 0.05) |

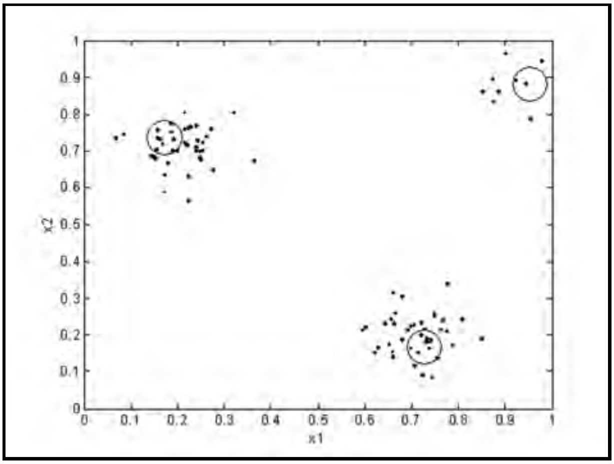


**Fig. 8.** Data set with 2 clusters each one comprising 40 patterns and an additional cluster with 10 patterns. The unfilled circles mark the prototypes.

### Example 2

The data set presented here is formed by three visually separable clusters. Two of them are composed of forty patterns whereas the third one is composed of only ten elements. Table 2 presents the details of the distribution.

**Table 2.** Details of the normal distribution of the data set of Example 2 (3 clusters)

| Cluster | #Points | Mean | Standard Deviation |
|---|---|---|---|
| 1 | 40 | (0.2; 0.7) | (0.05; 0.05) |
| 2 | 40 | (0.7; 0.2) | (0.05; 0.05) |
| 3 | 10 | (0.9; 0.9) | (0.05; 0.05) |

Figure 8 presents the data set and the means of the cluster prototypes for the ten distinct runs. The algorithm was able to recognize the correct number of clusters.

In the next experiment the cluster was further reduced to five points. Even for this more demanding case, the algorithm was able to recognize the data structure, see Fig. 9.

Next, the third "cluster" was further shrunk to only two points, see Fig. 10. In this case the algorithm negotiated these as outliers.

In order to infer the robustness of the algorithm to noise, we kept the 2 main clusters and injected 50 noise points uniformly distributed in the unit square,

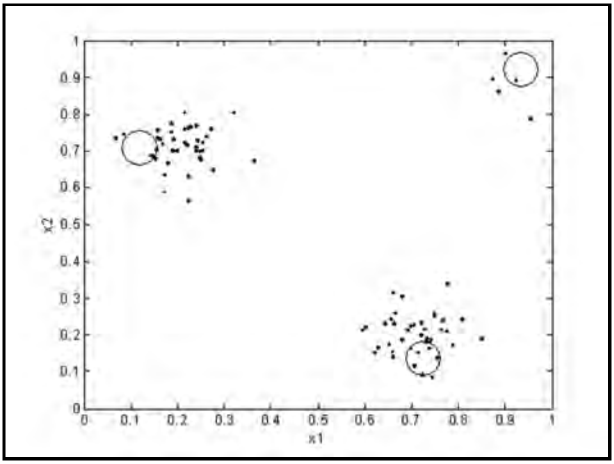


**Fig. 9.** Data set with 2 clusters each one comprising 40 patterns and an additional cluster with 5 patterns. The unfilled circles mark the prototypes.

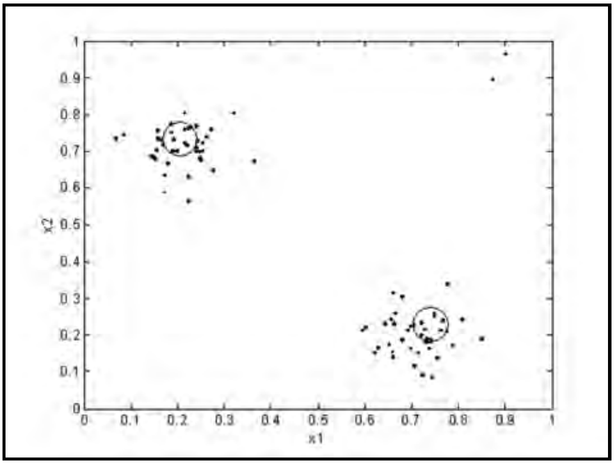


**Fig. 10.** Data set with 2 clusters each one comprising 40 patterns and two outliers. The unfilled circles mark the prototypes.

Fig. 11. Once again the algorithm was able to correctly identify the two clusters and find the optimal number of clusters.

### Ruspini Data Set

The Ruspini data set [68] has become a benchmark to assessing the performance of clustering algorithms. Figure 12 shows the Ruspini data set and clusters found by the proposed algorithm.

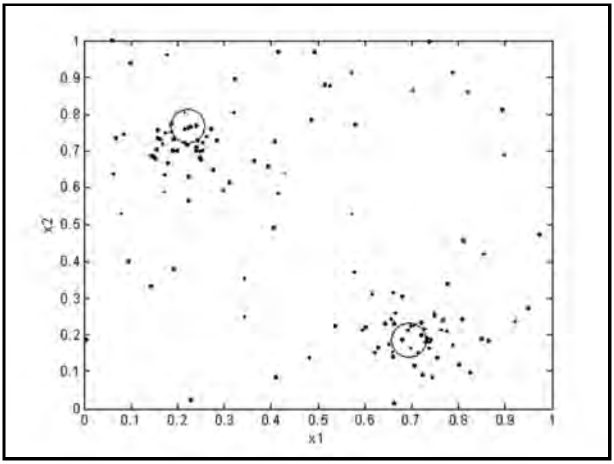


**Fig. 11.** Data set with 2 clusters each one comprising 40 patterns plus 50 noisy patterns. The unfilled circles mark the prototypes.

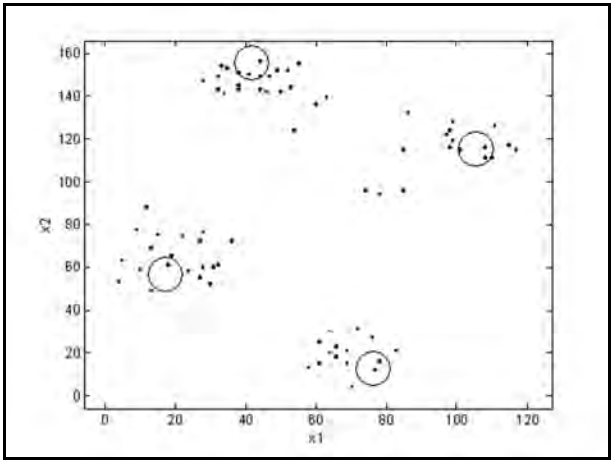


**Fig. 12.** Ruspini data set. The unfilled circles mark the prototypes.

Although the data has a much different scale than the unit square considered in the previous examples no additional adjustments were made to the algorithm. However this does not mean that when faced with huge data sets with large dimensionality these are not due. As a matter of fact for these cases a scale invariant distance function should be used and the relative weight of the distinguishability constraint should have to be analyzed. It is interesting to note that the algorithm was able to find the optimal number of clusters.

### Iris Plants

The Iris Plants database is one of the best known databases in the pattern recognition literature. The data set contains 3 classes referring to 3 types of iris plant physically labeled as Setosa, Versicolour and Virginica. There are 50 instances per class, which are described by 4 attributes: sepal length, sepal width, petal length and petal width. The first class is linearly separable from the others, but the other two classes are known to overlap each other in their numeric representations.

In this case the results of the algorithm were not conclusive. In exactly 50% of the runs the suggested number of clusters was 3 whereas in the remaining ones it produced only two clusters. This is a curious result since, although Iris database contains three physical labeled classes, from a geometrical point of view it is probably composed of only two distinct clusters. For the runs which identify 3 clusters the overall mean value of correct classifications was $92.75 \pm 1.58$. The error rate results exclusively from patterns misclassified in the two overlapping classes.

As a concluding remark it is worthwhile to stress that the translation of the human-defined semantic constraints referring to coverage and distinguishability of linguistic terms into the clustering framework is encouraging since in the presented experiments the proposed evolutionary algorithm, with the inclusion of the semantic constraints, was able to find the centers of the clusters and, more important, to determine the correct number of clusters without appealing to the computation of any kind of validity measures.

## 5 Conclusions

The fuzzy clustering framework provides a collection of tools well suited to discover structural information among challenging data sets with overlapping chunks of data and vaguely defined boundaries between clusters. Moreover, due to the wealthy of tools available and seamless integration of linguistic meaning it positions itself as an excellent mechanism to support the construction of information granules. These features assume a level of paramount importance not only for the data analyst, but also for the end-user who definitely needs a user-friendly mean to interact with the system. This interaction subsumes two facets: the human-like intuitive presentation of the clustering findings and consequent validation of these, as well as the transmission of additional knowledge that can be used to improve the clustering results. Along the first two parts of this chapter it was made clear that the fuzzy clustering framework is perfectly able to provide answers to each one of these aspects.

The leitmotif of the last part of the chapter was the discussion of the fuzzy clustering from a diametrically different point of view: *“Can the clustering process be conceptualized in a more human-oriented way?”* In this regard it was shown that the adoption of a set of semantic constraints aiming at the enhancement of the human perception of the system can also be applied to fuzzy clustering algorithms.

## References

1. Anderson, E.: The Irises of the Gaspe peninsula. Bull. Amer. Iris Soc. 59, 2–5 (1935)
2. Bargiela, A., Pedrycz, W.: Recursive Information Granulation: Aggregation and Interpretation Issues. IEEE Trans. on Systems, Man, and Cybernetics 33(1), 96–112 (2003)
3. Bargiela, A., Pedrycz, W.: A Model of Granular Data: a Design Problem with the Tchebyshev FCM. Soft Comput. 9, 155–163 (2005)
4. Bargiela, A., Pedrycz, W.: The roots of granular computing. In: Proc. Int. Conference on Granular Computing, pp. 806–809 (2006)
5. Bensaid, A., et al.: Partially supervised clustering for image segmentation. Pattern Recognition 29, 859–871 (1996)
6. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
7. Bezdek, J., Coray, C., Gunderson, R., Watson, J.: Detection and characterization of cluster substructure, I linear structure: Fuzzy c-lines. J. Appl. Math. 40(2), 339–357 (1981)
8. Bezdek, J., Coray, C., Gunderson, R., Watson, J.: Detection and characterization of cluster substructure, II Fuzzy c-varieties and convex combinations thereof. J. Appl. Math. 40(2), 358–372 (1981)
9. Bezdek, J., Hathaway, R., Sabin, M., Tucker, W.: Convergence theory for fuzzy c-means: Counterexamples and repairs. IEEE Trans. Syst., Man and Cybern. 17, 873–877 (1987)
10. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Accuracy Improvements in Linguistic Fuzzy Modeling. Physica-Verlag (2003)
11. Casillas, J., Cordón, O., Herrera, F., Magdalena, L. (eds.): Interpretability Issues in Fuzzy Modeling. Physica-Verlag (2003)
12. Castellano, G., Fanelli, A., Mencar, C.: Generation of interpretable fuzzy granules by a double-clustering technique. Archives of Control Sciences - Special Issue on Granular Computing 12(4), 397–410 (2002)
13. Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic Algorithms and Evolutionary Computation Series, vol. 5. Springer, Heidelberg (2002)
14. Dave, R.: Characterization and detection of noise in clustering. Pattern Recognition Letters 12, 657–664 (1991)
15. Dave, R., Sen, S.: On generalising the noise clustering algorithms. In: Proc. of the 7th IFSA World Congress, IFSA 1997, pp. 205–210 (1997)
16. Dave, R.: Fuzzy shell clustering and application to circle detection in digital images. Int. J. General Syst. 16, 343–355 (1990)
17. Davis, L.: Handbook of genetic algorithms. Van Nostrand Reinold (1991)
18. Egan, M.: Locating clusters in noisy data: a genetic fuzzy c-means clustering algorithm. In: Proceedings of the Conference of the North American Fuzzy Information Processing Society, pp. 178–182. IEEE, Los Alamitos (1998)
19. Egan, M., Krishnamoorthy, M., Rajan, K.: Comparative Study of a Genetic Fuzzy c-means Algorithm and a Validity Guided Fuzzy c-means Algorithm for Locating Clusters in Noisy Data. In: Proc. International Conference on Evolutionary Computation, pp. 440–445 (1998)
20. Eshelman, L., Schaffer, J.: Real-coded genetic algorithms and interval-schemata. In: Whitley, L.D. (ed.) Foundations of Genetic Algorithms 2, pp. 187–202. Morgan Kaufmann, San Mateo (1993)

21. Fazendeiro, P., Valente de Oliveira, J.: A Survey of Fuzzy Control Strategies for Neuromuscular Blockade Using Continuous Infusion of Atracurium. In: Proc. FUZZ-IEEE 2002, Honolulu, USA, pp. 547–552 (2002)
22. Fazendeiro, P., Valente de Oliveira, J.: A Working Hypothesis on the Semantics/Accuracy Synergy. In: Proc. Joint EUSFLAT-LFA, Barcelona, Spain, pp. 266–271 (2005)
23. Fazendeiro, P., Valente de Oliveira, J.: A semantic driven evolutive fuzzy clustering algorithm. In: Proc. of FUZZ-IEEE, London, UK, July 2007, pp. 1977–1982 (2007)
24. Fazendeiro, P., Valente de Oliveira, J., Pedrycz, W.: Multi-objective evolutionary optimization of accuracy and interpretability for neuromuscular blockade control. In: Proc. of IFSA 2005 - The International Fuzzy Systems Association World Congress, Beijing, China, pp. 908–913 (2005)
25. Fazendeiro, P., Valente de Oliveira, J., Pedrycz, W.: A Multi-Objective Design of a Patient and Anaesthetist-Friendly Neuromuscular Blockade Controller. IEEE Trans. on BioMedical Engineering 54(9), 1667–1678 (2007)
26. Forgy, E.: Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. Biometrics 21, 768–780 (1965)
27. Fralley, C., Raftery, A.: How many Clusters? Which Clustering Method? Answers via Model-Based Cluster Analysis. The Computer Journal 41(8), 578–588 (1998)
28. Frigui, H., Krishnapuram, R.: Clustering by competitive agglomeration. Pattern Recognition 30(7), 1109–1119 (1997)
29. Frigui, H., Krishnapuram, R.: A robust competitive clustering algorithm with applications in computer vision. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(5), 450–465 (1999)
30. Gath, I., Geva, A.: Unsupervised Optimal Fuzzy Clustering. IEEE Trans. Pattern Analysis and Machine Intelligence 11, 773–781 (1989)
31. Geva, A.: Hierarchical Unsupervised Fuzzy Clustering. IEEE Trans. Fuzzy Systems 7(6), 723–733 (1999)
32. Gustafson, D., Kessel, W.: Fuzzy clustering with a fuzzy covariance matrix. In: Proc. IEEE Conf. Decision Contr., San Diego, CA, pp. 761–766 (1979)
33. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On Clustering Validation Techniques. Journal of Intelligent Information Systems 17(2/3), 107–145 (2001)
34. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering algorithms and validity measures. In: Proc. 13th International Conference on Scientific and Statistical Database Management, pp. 3–22 (2001)
35. Hall, L., Ozyurt, B., Bezdek, J.: Clustering with a genetically optimized approach. IEEE Trans. on Evolutionary Computation 3(2), 103–112 (1999)
36. Hathaway, R., Bezdek, J.: Switching regression models and fuzzy clustering. IEEE Trans. Fuzzy Syst. 1, 195–204 (1993)
37. Hathaway, R., Bezdek, J.: Clustering incomplete relational data using the non-Euclidean relational fuzzy c-means algorithm. Pattern Recognition Letters 23, 151–160 (2002)
38. Hathaway, R., Bezdek, J., Hu, Y.: Generalized Fuzzy c-Means Clustering Strategies Using $L_p$ Norm Distances. IEEE Transactions on Fuzzy Systems 5, 576–582 (2000)
39. Hirota, K., Pedrycz, W.: Fuzzy computing for data mining. Proceedings of the IEEE 87(9), 1575–1600 (1999)
40. F. Hoppner, "Fuzzy shell clustering algorithms in image processing– Fuzzy c-rectangular and two rectangular shells," IEEE Trans. Fuzzy Syst., 5, 599–613 (1997)
41. Hoppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis. John Wiley, Chichester (1999)

42. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. ACM Comput. Surv. 31(3), 264–323 (1999)
43. Klawonn, F., Keller, A.: Fuzzy clustering with evolutionary algorithms. International Journal of Intelligent Systems 13(10/11), 975–991 (1998)
44. Klawonn, F., Höppner, F.: What is fuzzy about fuzzy clustering? Understanding and improving the concept of the fuzzifier. In: Berthold, M.R., Lenz, H.-J., Bradley, E., Kruse, R., Borgelt, C. (eds.) IDA 2003. LNCS, vol. 2810, pp. 254–264. Springer, Heidelberg (2003)
45. Krisnapuram, R., Frigui, H., Nasroui, O.: Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation. IEEE Trans. Fuzzy Syst. 3(1), 29–60 (1995)
46. Krishnapuram, R., Joshi, A., Yi, L.: A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering. In: Proc. IEEE International Fuzzy Systems Conference, pp. 1281–1286 (1999)
47. Krishnapuram, R., Keller, J.: A possibilistic approach to clustering. IEEE Trans. on Fuzzy Systems 1, 98–110 (1993)
48. Krishnapuram, R., Keller, J.: The possibilistic C-Means algorithm: insights and recommendations. IEEE Trans. on Fuzzy Systems 4, 385–393 (1996)
49. Liu, G.: Introduction to Combinatorial Mathematics. Mc-Graw-Hill, New York (1968)
50. Liu, H., Huang, S.-T.: Evolutionary semi-supervised fuzzy clustering. Pattern Recognition Letters 24, 3105–3113 (2003)
51. Liu, J., Xie, W.: A genetics-based approach to fuzzy clustering. In: Proc. of the IEEE International Conference on Fuzzy Systems, vol. 4, pp. 2233–2240 (1995)
52. Loia, V., Pedrycz, W., Senatore, S.: P-FCM: a proximity-based fuzzy clustering for user-centered web applications. International Journal of Approximate Reasoning 34, 121–144 (2003)
53. Lorette, A., Descombes, X., Zerubia, J.: Fully unsupervised fuzzy clustering with entropy criterion. Proceedings 15th International Conference on Pattern Recognition 3, 986–989 (2000)
54. MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. 1, 281–297 (1967)
55. Maulik, U., Bandyopadhyay, S.: Performance evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(12), 1650–1654 (2002)
56. Maulik, U., Bandyopadhyay, S.: Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. IEEE Trans. on Geoscience and Remote Sensing 41(5), 1075–1081 (2003)
57. Miller, G.: The magic number seven, plus or minus two: Some limits on our capacity for processing information. Psychol. Rev. 63, 81–97 (1956)
58. Milligan, G., Cooper, C.: An examination of procedures for Determining the Number of Clusters in a Data Set. Psychometrika 50(2), 159–179 (1985)
59. Ohashi, Y.: Fuzzy Clustering and robust estimation. In: Proc. of 9th Meeting SAS User Group Int., Florida (1984)
60. Pal, N., Bezdek, J.: On cluster validity for the fuzzy c-means model. IEEE Transactions on Fuzzy Systems 3(3), 370–379 (1995)
61. Pedrycz, W.: Conditional fuzzy clustering in the design of radial basis function neural network. IEEE Trans. Neural Networks 9, 601–612 (1998)
62. Pedrycz, W., Valente de Oliveira, J.: An Algorithmic Framework for Development and Optimization of Fuzzy Models. Special issue of Fuzzy Sets and Systems on Fuzzy Modelling 80(1), 37–55 (1996)

63. Pedrycz, W., Vukovich, G.: Abstraction and Specialization of Information Granules. IEEE Trans. on Systems, Man and Cybernetics, Part B 31(1), 106–111 (2001)
64. Pedrycz, W., Waletzky, J.: Fuzzy Clustering with Partial Supervision. IEEE Trans. on Systems, Man and Cybernetics, Part B 27(5), 787–795 (1997)
65. Runkler, T., Bezdek, J.: Alternating cluster estimation: A new tool for clustering and function approximation. IEEE Trans. Fuzzy Syst. 7, 377–393 (1999)
66. Runkler, T.: Ant colony optimization of clustering models. International Journal of Intelligent Systems 20(12), 1233–1261 (2005)
67. Runkler, T., Katz, C.: Fuzzy Clustering by Particle Swarm Optimization. In: Proc. IEEE International Conference on Fuzzy Systems, pp. 601–608 (2006)
68. Ruspini, E.: A New Approach to Clustering. Information Control 15, 22–32 (1969)
69. Sarkar, M.: Evolutionary programming-based fuzzy clustering. In: Proc. of the Fifth Annual Conference on Evolutionary Programming, pp. 247–254. MIT Press, Cambridge (1996)
70. Valente de Oliveira, J.: A design methodology for fuzzy system interfaces. IEEE Trans. on Fuzzy Systems 3(4), 404–414 (1995)
71. Valente de Oliveira, J.: Semantic Constraints for Membership Function Optimization. IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Man 29(1), 128–138 (1999)
72. Valente de Oliveira, J., Pedrycz, W. (eds.): Advances in Fuzzy Clustering and Its Applications. Wiley, Chichester (2007)
73. Wei, C.-H., Fahn, C.-S.: A distributed approach to fuzzy clustering by genetic algorithms. In: Proceedings of the Asian Fuzzy Systems Symposium, pp. 350–357 (1996)
74. Xie, X., Beni, G.: A Validity Measure for Fuzzy Clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 13, 841–847 (1991)
75. Xu, R., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Trans. on Neural Networks 16(3), 645–678 (2005)
76. Yuan, B., Klir, G., Swan-Stone, J.: Evolutionary fuzzy c-means clustering algorithm. In: Proc. of IEEE International Conference on Fuzzy Systems, vol. 4, pp. 2221–2226 (1995)
77. Zadeh, L.: Fuzzy sets. Inform. Contr. 8, 338–353 (1965)
78. Zadeh, L.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. on Systems, Man, and Cybernetics SMC-3(1), 28–44 (1973)
79. Zadeh, L.: The concept of linguistic variable and its application to approximate reasoning. Information Sciences 8, 199–249 (part I) (1975); 301–357 (part II)
80. Zadeh, L.: The concept of linguistic variable and its application to approximate reasoning. Information Sciences 9, 43–80 (1976) (part III)
81. Zadeh, L.: Fuzzy logic = computing with words. IEEE Trans. on Fuzzy Systems 4(2), 103–111 (1996)
82. Zadeh, L.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–127 (1997)
83. Zadeh, L.: Some reflections on soft computing, granular computing and their roles in the conception, design and utilization of information/intelligent systems. Soft Computing 2(1), 23–25 (1998)

# Many-Valued Logic Tools for Granular Modeling

Siegfried Gottwald

Leipzig University, Institute for Logic and Philosophy of Science, Beethovenstr. 15,
04107 Leipzig, Germany
gottwald@uni-leipzig.de

**Abstract.** A core tool for granular modeling is the use of linguistic rules, e.g. in fuzzy control approaches. We provide the reader with basic mathematical tools to discuss the behavior of system of such linguistic rules.

These mathematical tools range from fuzzy logic and fuzzy set theory, through the consideration of fuzzy relation equations, up to discussions of interpolation strategies and to the use of aggregation operators.

## 1 Introduction

In their everyday behavior humans quite often reason qualitatively. And they do this rather successful, even in handling complex situations. For the knowledge engineer it is an interesting and important challenge to adopt this type of behavior in his modeling activities.

Fuzzy sets and fuzzy logic have been designed and developed just for this purpose over the last decades. And they offer a rich variety of methods for this purpose: methods, which have quite different mathematical tools as origin and background.

In this chapter we concentrate on tools which are related to logic in the formal mathematical sense of the word, and to a set theoretic – particularly a relation theoretic – background.

The machinery of mathematical fuzzy logic, which is the topic of the Sections 2 and 3, offers a background to model and provides a tool to understand the treatment of vague information.

The machinery of fuzzy relation equations, discussed in Sections 4 to 6, is a core tool from the mathematical background for the understanding of fuzzy control approaches, and a prototypical case for the use of information granules.

All of these tools derive a lot of flexibility from the use of t-norms, i.e. of binary operations in the unit interval which are associative, commutative, and isotonic, and which have 1 as their neutral element.

Some further reflections on the topic of fuzzy relation equations provide additional mathematical tools to understand them: the interpolation method, as well as the use of aggregation operators.

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 153–184.
springerlink.com

# 2 Fuzzy Sets and Many-Valued Logic

## 2.1 Membership Degrees as Truth Degrees

A *fuzzy set* $A$ is characterized by its generalized characteristic function $\mu_A : \mathbb{X} \to [0, 1]$, called *membership function* of $A$ and defined over some given *universe of discourse* $\mathbb{X}$, i.e. it is a *fuzzy subset* of $\mathbb{X}$.

The essential idea behind this approach was to have the membership degree $\mu_A(a)$ for each point $a \in \mathbb{X}$ as a gradation of its membership with respect to the fuzzy set $A$. And this degree just is a degree to which the sentence *"a is a member of A"* holds true. Hence it is natural to interpret the membership degrees of fuzzy sets as truth degrees of the membership predicate in some (suitable system of) many-valued logic $\mathsf{S}$.

To do this in a reasonable way one has to accept some minimal conditions concerning the formal language $\mathcal{L}_\mathsf{S}$ of this system.

Disregarding – for simplicity – fuzzy sets of type 2 and of every higher type as well one has, from the set theoretic point of view, fuzzy sets as (generalized) sets of first level over a given class of urelements, the universe of discourse for these fuzzy sets. Therefore the intended language needs besides a (generalized, i.e. graded) binary membership predicate $\varepsilon$ e.g. two types of variables: (i) lower case latin letters $a, b, c, \ldots, x, y, z$ for urelements, i.e. for points of the universe of discourse $\mathbb{X}$, and (ii) upper case latin letters $A, B, C, \ldots$ for fuzzy subsets of $\mathbb{X}$. And of course it has some set of connectives and some quantifiers – and thus a suitable notion of well-formed formula.

Having in mind the standard fuzzy sets with membership degrees in the real unit interval $[0, 1]$ thus forces to assume that $\mathsf{S}$ is an infinitely many-valued logic.

It is not necessary to fix all the details of the language $\mathcal{L}_\mathsf{S}$ in advance. We suppose, for simplicity of notation, that from the context it shall always be clear which objects the individual symbols are to denote.[1] Denoting the truth degree of a well-formed formula $H$ by $[\![H]\!]$, to identify membership degrees with suitable truth degrees then means to put

$$\mu_A(x) = [\![\, x \,\varepsilon\, A \,]\!]. \tag{1}$$

This type of interpretation proves quite useful: it opens the doors to clarify far reaching analogies between notions and results related to fuzzy sets and those ones related to usual sets, cf. [14, 16].

## 2.2 Doing Fuzzy Set Theory Using MVL Language

Based on the main idea to look at the membership degrees of fuzzy sets as truth degrees of a suitable membership predicate, it e.g. becomes quite natural to describe fuzzy sets by a (generalized) class term notation, adapting the corresponding notation $\{x \mid H(x)\}$ from traditional set theory and introducing a corresponding notation for fuzzy sets by

[1] Which, formally, means that we assume that a valuation always is determined by the context and has not explicitly to be mentioned.

$$A = \{x \in \mathbb{X} \parallel H(x)\} \;\Leftrightarrow_{\text{def}}\; \mu_A(x) = [\![H(x)]\!] \quad \text{for all } x \in \mathbb{X}, \tag{2}$$

with now $H$ a well-formed formula of the language $\mathcal{L}_{\mathsf{S}}$. As usual, the shorter notation $\{x \parallel H(x)\}$ is also used, and even preferred.

### 2.2.1 Fuzzy Set Algebra

With this notation the intersection and the cartesian product of fuzzy sets $A, B$ are, even in their t-norm based version, determined as

$$\begin{aligned} A \cap_{\boldsymbol{t}} B &= \{x \parallel x \,\varepsilon\, A \wedge_{\boldsymbol{t}} x \,\varepsilon\, B\}, \\ A \times_{\boldsymbol{t}} B &= \{(x,y) \parallel x \,\varepsilon\, A \wedge_{\boldsymbol{t}} y \,\varepsilon\, B\}, \end{aligned}$$

and the standard, i.e. min-based form of the *compositional rule of inference*[2] (CRI for short) applied w.r.t. a fuzzy relation $R$ and a fuzzy set $A$ becomes

$$A \circ R = R''A = \{y \parallel \exists x (x \,\varepsilon\, A \wedge_{\boldsymbol{t}} (x,y) \,\varepsilon\, R)\} \tag{3}$$

with $\boldsymbol{t} = \min$. This is the analogue of a formula well known from elementary relation algebra which describes the full image of a set $A$ under a relation $R$.

Of course, also other choices of the t-norm involved here are possible.

Also for the inclusion relation between fuzzy sets this approach works well. The standard definition of inclusion amounts to

$$A \subset B \quad \Leftrightarrow \quad \mu_A(x) \leq \mu_B(x) \;\text{ for all }\; x \in \mathbb{X}$$

which in the language of many-valued logic is the same as

$$A \subset B \quad \Leftrightarrow \quad \models \forall x (x \,\varepsilon\, A \rightarrow_{\boldsymbol{t}} x \,\varepsilon\, B) \tag{4}$$

w.r.t. any one R-implication connective based on a left continuous t-norm.

Obviously this version (4) of inclusion is easily generalized to a “fuzzified”, i.e. (truly) many-valued inclusion relation defined as

$$A \subseteqq B =_{\text{def}} \forall x (x \,\varepsilon\, A \rightarrow_{\boldsymbol{t}} x \,\varepsilon\, B). \tag{5}$$

And this many-valued inclusion relation for fuzzy sets has still nice properties, e.g. it is $\boldsymbol{t}$-transitive, i.e. one has:

$$\models \; (A \subseteqq B \wedge_{\boldsymbol{t}} B \subseteqq C \rightarrow_{\boldsymbol{t}} A \subseteqq C).$$

### 2.2.2 Fuzzy Relation Theory

This natural approach (3) toward the compositional rule of inference is almost the same as the usual definition of the relational product $R \circ S$ of two fuzzy relation, now – even related to some suitable t-norm – to be determined as

[2] This compositional rule of inference is of central importance for the applications of fuzzy sets to fuzzy control and to approximate reasoning, cf. Chapter 1 of the Handbook volume “Fuzzy Sets in Approximate Reasoning and Information Systems” edited by J. Bezdek, D. Dubois and H. Prade.

$$R \circ_{\boldsymbol{t}} S = \{(x,y) \parallel \exists z\big((x,z)\,\varepsilon\, R \wedge_{\boldsymbol{t}} (z,y)\,\varepsilon\, S\big)\}.$$

Relation properties become, in this context, again characterizations which formally read as the corresponding properties of crisp sets. Consider, as an example, transitivity of a fuzzy (binary) relation $R$ in the universe of discourse $\mathbb{X}$ w.r.t. some given t-norm. The usual condition for all $x, y, z \in \mathbb{X}$

$$\boldsymbol{t}(\mu_R(x,y), \mu_R(y,z)) \leq \mu_R(x,z)$$

in the language $\mathcal{L}_{\mathsf{S}}$ of the intended suitable $[0,1]$-valued system for fuzzy set theory becomes the condition

$$\models R(x,y) \wedge_{\boldsymbol{t}} R(y,z) \rightarrow_{\boldsymbol{t}} R(x,z)$$

for all $x, y, z \in \mathbb{X}$ or, even better, becomes

$$\models \forall xyz\big(R(x,y) \wedge_{\boldsymbol{t}} R(y,z) \rightarrow_{\boldsymbol{t}} R(x,z)\big) \tag{6}$$

with the universal quantifier $\forall$ as in the Łukasiewicz systems.

This point of view not only opens the way for a treatment of fuzzy relations quite analogous to the usual discussion of properties of crisp relations, it also opens the way to consider *graded versions* of properties of fuzzy relations, cf. [14]. In the case of transitivity, a graded or "fuzzified" predicate Trans with the intended meaning "is transitive" may be defined as

$$\mathsf{Trans}(R) =_{\text{def}} \forall xyz\big(R(x,y) \wedge_{\boldsymbol{t}} R(y,z) \rightarrow_{\boldsymbol{t}} R(x,z)\big). \tag{7}$$

This point of view has recently been treated in more detail e.g. in [1].

# 3 T-Norm-Based Mathematical Fuzzy Logics

## 3.1 Basic Infinite Valued Logics

If one looks for infinite valued logics of the kind which is needed as the underlying logic for a theory of fuzzy sets, one finds three main systems:

- the Łukasiewicz logic $\mathsf{L}$ as explained in [31];
- the Gödel logic $\mathsf{G}$ from [11];
- the product logic $\mathit{\Pi}$ studied in [23].

In their original presentations, these logics look rather different, regarding their propositional parts. For the first order extensions, however, there is a unique strategy: one adds a universal and an existential quantifier such that quantified formulas get, respectively, as their truth degrees the infimum and the supremum of all the particular cases in the range of the quantifiers.

As a reference for these and also other many-valued logics in general, the reader may consult [16].

### 3.1.1 Gödel Logic

The simplest one of these logics is the *Gödel logic* $\mathsf{G}$ which has a conjunction $\wedge$ and a disjunction $\vee$ defined by the minimum and the maximum, respectively, of the truth degrees of the constituents:

$$u \wedge v = \min\{u, v\}\,, \qquad u \vee v = \max\{u, v\}\,. \tag{8}$$

For simplicity we denote here and later on the connectives and the corresponding truth degree functions by the same symbol.

The Gödel logic has also a negation $\sim$ and an implication $\rightarrow_{\mathsf{G}}$ defined by the truth degree functions

$$\sim u = \begin{cases} 1\,, \text{ if } \ u = 0; \\ 0\,, \text{ if } \ u > 0. \end{cases} \qquad u \rightarrow_{\mathsf{G}} v = \begin{cases} 1\,, \text{ if } \ u \le v; \\ v\,, \text{ if } \ u > v. \end{cases} \tag{9}$$

### 3.1.2 Łukasiewicz Logic

The *Łukasiewicz logic* $\mathsf{L}$ was originally designed in [31] with only two primitive connectives, an implication $\rightarrow_{\mathsf{L}}$ and a negation $\neg$ characterized by the truth degree functions

$$\neg u = 1 - u\ , \qquad u \rightarrow_{\mathsf{L}} v = \min\{1, 1 - u + v\}\,. \tag{10}$$

However, it is possible to define further connectives from these primitive ones. With

$$\varphi \,\&\, \psi =_{\mathrm{df}} \neg(\varphi \rightarrow_{\mathsf{L}} \neg\psi)\ , \qquad \varphi \veebar \psi =_{\mathrm{df}} \neg\varphi \rightarrow_{\mathsf{L}} \psi \tag{11}$$

one gets a (strong) conjunction and a (strong) disjunction with truth degree functions

$$u \,\&\, v = \max\{u + v - 1, 0\}\,, \qquad u \veebar v = \min\{u + v, 1\}\,, \tag{12}$$

usually called the *Łukasiewicz* (arithmetical) *conjunction* and the *Łukasiewicz* (arithmetical) *disjunction*. It should be mentioned that these connectives are linked together via a De Morgan law using the standard negation of this system:

$$\neg(u \,\&\, v) = \neg u \veebar \neg v\,. \tag{13}$$

With the additional definitions

$$\varphi \wedge \psi =_{\mathrm{df}} \varphi \,\&\, (\varphi \rightarrow_{\mathsf{L}} \psi) \qquad \varphi \vee \psi =_{\mathrm{df}} (\varphi \rightarrow_{\mathsf{L}} \psi) \rightarrow_{\mathsf{L}} \psi \tag{14}$$

one gets another (weak) conjunction $\wedge$ with truth degree function min, and a further (weak) disjunction $\vee$ with max as truth degree function, i.e. one has the conjunction and the disjunction of the Gödel logic also available.

### 3.1.3 Product Logic

The *product logic $\Pi$*, in detail explained in [23], has a fundamental conjunction $\odot$ with the ordinary product of reals as its truth degree function, as well as an implication $\rightarrow_{\Pi}$ with truth degree function

$$u \rightarrow_{\Pi} v = \begin{cases} 1\,, & \text{if} \;\; u \leq v; \\ \frac{u}{v}\,, & \text{if} \;\; u < v. \end{cases} \tag{15}$$

Additionally it has a truth degree constant $\overline{0}$ to denote the truth degree zero.

In this context, a negation and a further conjunction are defined as

$$\sim \varphi =_{\mathrm{df}} \varphi \rightarrow_{\Pi} \overline{0}\,, \qquad \varphi \wedge \psi =_{\mathrm{df}} \varphi \odot (\varphi \rightarrow_{\Pi} \psi)\,. \tag{16}$$

Routine calculations show that both connectives coincide with the corresponding ones of the Gödel logic. And also the disjunction $\vee$ of the Gödel logic becomes available, now via the definition

$$\varphi \vee \psi =_{\mathrm{df}} ((\varphi \rightarrow_{\Pi} \psi) \rightarrow_{\Pi} \psi) \wedge ((\psi \rightarrow_{\Pi} \varphi) \rightarrow_{\Pi} \varphi)\,. \tag{17}$$

### 3.2 Standard and Algebraic Semantics

These fundamental infinite valued logics have their *standard semantics* as explained: the real unit interval $[0, 1]$ as truth degree set, and the connectives (and quantifiers) as mentioned.

In the standard way, as known from classical logic, one then can introduce for each formula $\varphi$ the notion of its *validity in a model*, which in these logics means that $\varphi$ has the truth degree 1 w.r.t. this model. By a *model* we mean either—in the propositional case—an evaluation of the propositional variables by truth degrees, or—in the first-order case—a suitable interpretation of all the non-logical constants together with an assignment of the variables.

Based upon this, one defines *logical validity* of a formula $\varphi$ as validity of $\varphi$ in each model, and the *entailment* relation holds between a set $\Sigma$ of formulas and a formula $\varphi$ iff each model of $\Sigma$ is also a model of $\varphi$.

In the standard terminology of many-valued logic in general, this means that all the three systems $\mathsf{G}, \mathsf{L}, \Pi$ have the truth degree one as their only designated truth degree.

Besides these standard semantics, all three of these basic infinite valued logics have also *algebraic semantics* determined by suitable classes $\mathcal{K}$ of truth degree structures. The situation is similar here to the case of classical logic: the logically valid formulas in classical logic are also just all those formulas which are valid in all Boolean algebras.

Of course, these structures should have the same signature as the language $\mathcal{L}$ of the corresponding logic. This means that these structures provide for each connective of the language $\mathcal{L}$ an operation of the same arity, and they have to have—in the case that one discusses the corresponding first order logics—suprema and infima for all those subsets which may appear as value sets of formulas. Particularly, hence, they have to be (partially) ordered, or at least pre-ordered.

For each formula $\varphi$ of the language $\mathcal{L}$ of the corresponding logic, for each such structure $\mathbf{A}$, and for each evaluation $e$ which maps the set of propositional variables of $\mathcal{L}$ into the carrier of $\mathbf{A}$, one has to define a value $e(\varphi)$, and finally one has to define what it means that such a formula $\varphi$ is *valid in* $\mathbf{A}$. Then a formula $\varphi$ is *logically valid* w.r.t. this class $\mathcal{K}$ iff $\varphi$ is valid in all structures from $\mathcal{K}$.

The standard way to arrive at such classes of structures is to start from the Lindenbaum algebra of the corresponding logic, i.e. its algebra of formulas modulo the congruence relation of logical equivalence. For this Lindenbaum algebra one then has to determine a class of similar algebraic structures which—ideally—forms a variety.

For the Gödel logic such a class of structures is, according to the completeness proof of [8], the class of all *Heyting algebras*, i.e. of all relatively pseudo-complemented lattices, which satisfy the prelinearity condition

$$(u \rightarrowtail v) \sqcup (v \rightarrowtail u) = \mathbf{1}\,. \tag{18}$$

Here $\sqcup$ is the lattice join and $\rightarrowtail$ the relative pseudo-complement.

For the Łukasiewicz logic the corresponding class of structures is the class of all *MV-algebras*, first introduced again within a completeness proof in [3], and extensively studied in [5].

And for the product logic the authors of [23] introduce a class of lattice ordered semigroups which they call *product algebras*.

It is interesting to recognize that all these structures—pre-linear Heyting algebras, MV-algebras, and product algebras—are abelian lattice ordered semigroups with an additional "residuation" operation.

### 3.3 Logics with t-Norm Based Connectives

The fundamental infinite valued logics from Section 3.1 look quite different if one has in mind the form in which they first were presented.

Fortunately, however, there is a common generalization which allows to present all these three logics in a uniform way. In this uniform presentation one of the conjunction connectives becomes a core role: $\wedge$ in the system $\mathsf{G}$, & in the system $\mathsf{L}$, and $\odot$ in the system $\Pi$.

But this uniform generalization covers a much larger class of infinite valued logics over $[0, 1]$: the core conjunction connective—which shall now in general be denoted &—has only to have a truth degree function $\otimes$ which, as a binary operation in the real unit interval, should be an associative, commutative, and isotonic operation which has 1 as a neutral element, i.e. should satisfy for arbitrary $x, y, z \in [0, 1]$:

(T1) $x \otimes (y \otimes z) = (x \otimes y) \otimes z$,
(T2) $x \otimes y = y \otimes x$,
(T3) if $x \leq y$ then $x \otimes z \leq y \otimes z$,
(T4) $x \otimes 1 = x$.

Such binary operations are known as *t-norms* and have been used in the context of probabilistic metric spaces, cf. e.g. [29]. At the same time they are considered as natural candidates for truth degree functions of conjunction connectives. And from such a t-norm one is able to derive (essentially) all the other truth degree functions for further connectives.

The minimum operation $u \wedge v$ from (8), the Łukasiewicz arithmetic conjunction $u \mathbin{\&} v$ from (12), and the ordinary product are the best known examples of t-norms.

In algebraic terms, such a t-norm $\otimes$ makes the real unit interval into an *ordered monoid*, i.e. into an abelian semigroup with unit element. And this ordered monoid is even *integral*, i.e. its unit element is at the same time the universal upper bound of the ordering. Additionally this monoid has because of

$$0 \otimes x \leq 0 \otimes 1 = 0 \tag{19}$$

the number 0 as an *annihilator.*

Starting from a t-norm $\otimes$ one finds a truth degree function $\rightarrowtail$ for an implication connective via the *adjointness condition*

$$x \otimes z \leq y \quad \Longleftrightarrow \quad z \leq (x \rightarrowtail y)\,. \tag{20}$$

However, to guarantee that this adjointness condition (20) determines the operation $\rightarrowtail$ uniquely, one has to assume that the t-norm $\otimes$ is a *left continuous* function in both arguments. Indeed, the adjointness condition (20) is equivalent to the condition that $\otimes$ is left continuous in both arguments, cf. [16].

Instead of this adjointness condition (20) one could equivalently either give the direct definition

$$x \rightarrowtail y = \sup\{z \mid x \otimes z \leq y\} \tag{21}$$

of the *residuation operation* $\rightarrowtail$, or one could force the t-norm $\otimes$ to have the *sup-preservation property*

$$\sup_{i\to\infty} (x_i \otimes y) = (\sup_{i\to\infty} x_i) \otimes y \tag{22}$$

for each $y \in [0,1]$ and each non-decreasing sequence $(x_i)_{i\to\infty}$ from the real unit interval.

In this framework one additionally introduces a further unary operation $-$ by

$$-x =_{\mathrm{df}} x \rightarrowtail 0\,, \tag{23}$$

and considers this as the truth degree function of a negation connective. That this works also in the formalized language of the corresponding system of logic forces to introduce into this language a truth degree constant $\overline{0}$ to denote the truth degree zero.

And finally one likes to have the weak conjunction and disjunction connectives $\wedge, \vee$ available. These connectives should also be added to the vocabulary. However, it suffices to add only the min-conjunction $\wedge$, because then for each left continuous t-norm $\otimes$ and its residuated implication $\rightarrowtail$ one has, completely similar to the situation (17) in the product logic,

$$u \vee v = ((u \rightarrowtail v) \rightarrowtail v) \wedge ((v \rightarrowtail u) \rightarrowtail u)\,. \tag{24}$$

All these considerations lead in a natural way to algebraic structures which, starting from the unit interval, consider a left continuous t-norm $\otimes$ together with its residuation operation $\rightarrowtail$, with the minimum-operation $\wedge$, and the maximum operation $\vee$ as basic operations of such an algebraic structure, and with the particular truth degrees $0, 1$ as fixed objects (i.e. as nullary operations) of the structure. Such an algebraic structure

$$\langle [0,1], \wedge, \vee, \otimes, \rightarrowtail, 0, 1 \rangle \tag{25}$$

shall be coined to be a *t-norm algebra.*

### 3.4 Continuous t-Norms

Among the large class of all t-norms the continuous ones are best understood. A t-norm is continuous iff it is continuous as a real function of two variables, or equivalently, iff it is continuous in each argument (with the other one as a parameter), cf. [16, 29].

Furthermore, all continuous t-norms are ordinal sums of only three of them: the Łukasiewicz arithmetic t-norm $u \,\&\, v$ from (12), the ordinary product t-norm, and the minimum operation $u \wedge v$. The definition of an ordinal sum of t-norms is the following one.

**Definition 1.** *Suppose that* $([a_i, b_i])_{i \in I}$ *is a countable family of non-overlapping proper subintervals of the unit interval* $[0,1]$*, let* $(\boldsymbol{t}_i)_{i \in I}$ *be a family of t-norms, and let* $(\varphi_i)_{i \in I}$ *be a family of mappings such that each* $\varphi_i$ *is an order isomorphism from* $[a_i, b_i]$ *onto* $[0,1]$*. Then the* (*generalized*) ordinal sum *of the combined family* $(([a_i, b_i], \boldsymbol{t}_i, \varphi_i))_{i \in I}$ *is the binary function* $T : [0,1]^2 \to [0,1]$ *characterized by*

$$T(u,v) = \begin{cases} {\varphi_k}^{-1}(\boldsymbol{t}_k(\varphi_k(u), \varphi_k(v)), & \text{if } u, v \in [a_k, b_k] \\ \min\{u, v\} & \text{otherwise.} \end{cases} \tag{26}$$

It is easy to see that an order isomorphic copy of the minimum t-norm is again the minimum operation. Thus the whole construction of ordinal sums of t-norms even allows to assume that the summands are formed from t-norms different from the minimum t-norm. This detail, however, shall be inessential for the present considerations.

But it should be mentioned that all the endpoints $a_i, b_i$ of the interval family $([a_i, b_i])_{i \in I}$ give *idempotents* of the resulting ordinal sum t-norm $T$:

$$T(a_i, a_i) = a_i \,, \quad T(b_i, b_i) = b_i \quad \text{for all } i \in I.$$

Conversely, if one knows all the idempotents of a given continuous t-norm $\boldsymbol{t}$, i.e. all $u \in [0,1]$ with $\boldsymbol{t}(u,u) = u$, then one is able to give a representation of $\boldsymbol{t}$ as an ordinal sum, as explained again in [29].

The general result, given e.g. in [16, 29], reads as follows.

**Theorem 1.** *Each continuous t-norm* $\boldsymbol{t}$ *is the* (*generalized*) *ordinal sum of* (*isomorphic*) *copies of the Łukasiewicz t-norm, the product t-norm, and the minimum t-norm.*

As was mentioned in Section 3.1, the t-norm based logics which are determined by these three t-norms are well known and adequately axiomatized.

Therefore one is interested to find adequate axiomatizations also for further continuous t-norms. A global solution of this problem, i.e. a solution which did not only cover some few particular cases, appeared as quite difficult. Therefore, instead, one first has been interested to find all those formulas of the language of t-norm based systems which are logically valid in each one of these logics.

There seems to be a natural way to get an algebraic semantics for these considerations: the class of all t-norm algebras with a continuous t-norm should either form such an algebraic semantics, or should be a constitutive part—preferably a generating set—of a variety of algebraic structures which form such an algebraic semantics.

However, there seems to be an inadequacy in the description of this algebraic semantics: on the one hand the notion of t-norm algebra is a purely algebraic notion, the notion of continuity of a t-norm on the other hand is an analytical one. Fortunately, there is a possibility to give an algebraic characterization for the continuity of t-norms. It needs a further notion.

**Definition 2.** *A t-norm algebra* $\langle [0,1], \wedge, \vee, \otimes, \rightarrowtail, 0, 1 \rangle$ *is* divisible *iff one has for all* $a, b \in L$*:*

$$a \wedge b = a \otimes (a \rightarrowtail b)\,. \tag{27}$$

And this notion gives the algebraic counterpart for the continuity, as shown e.g. in [16, 29].

**Proposition 1.** *A t-norm algebra* $\langle [0,1], \wedge, \vee, \otimes, \rightarrowtail, 0, 1 \rangle$ *is divisible iff the t-norm* $\otimes$ *is continuous.*

### 3.5 The Logic of Continuous t-Norms

Instead of considering for each particular t-norm $\boldsymbol{t}$ the $\boldsymbol{t}$-based logic, it seems preferable and more interesting to consider the common logic of all t-norms of some kind. This was first realized for the class of all continuous t-norms by Hájek [22]. This logic should have as a natural *standard semantics* the class of all t-norm algebras with a continuous t-norm.

However, to built up a logic with an algebraic semantics determined by a class $\mathcal{K}$ of algebraic structures becomes quite natural in the cases that this class $\mathcal{K}$ is a variety: i.e. a class which is equationally definable—or equivalently, in more algebraic terms, which is closed under forming direct products, substructures, and homomorphic images.

Unfortunately, the class of t-norm algebras (with a continuous t-norm or not) is not a variety: it is not closed under direct products because each t-norm algebra is linearly ordered, but the direct products of linearly ordered structures are not linearly ordered, in general. Hence one may expect that it would be helpful for the development of a logic of continuous t-norms to extend the class of all divisible t-norm algebras in a moderate way to get a variety. And indeed this idea works, and is in detail explained in [22].

The core points are that one considers instead of the divisible t-norm algebras, which are linearly ordered integral monoids as mentioned previously, now lattice ordered integral monoids which are divisible, which have an additional residuation operation connected with the semigroup operation via an adjointness condition like (20), and which satisfy a pre-linearity condition like (18). These structures have been called BL-algebras; they are completely defined in the following way.

**Definition 3.** *A* BL-algebra $\mathbf{L} = \langle L, \vee, \wedge, *, \rightarrow, \mathbf{0}, \mathbf{1}\rangle$ *is an algebraic structure such that*

(*i*) $(L, \vee, \wedge, \mathbf{0}, \mathbf{1})$ *is a bounded lattice, i.e. has* $\mathbf{0}$ *and* $\mathbf{1}$ *as the universal lower and upper bounds w.r.t. the lattice ordering* $\leq$*,*

(*ii*) $(L, *, \mathbf{1})$ *is an abelian monoid, i.e. a commutative semigroup with unit* $\mathbf{1}$ *such that the multiplication* $*$ *is associative, commutative and satisfies* $\mathbf{1} * x = x$ *for all* $x \in L$*,*

(*iii*) *the binary operations* $*$ *and* $\rightarrow$ *form an adjoint pair, i.e. satisfy for all* $x, y, z \in L$ *the* adjointness *condition*

$$z \leq (x \rightarrow y) \iff x * z \leq y, \tag{28}$$

(*iv*) *and moreover, for all* $x, y \in L$ *one has satisfied the* pre-linearity *condition*

$$(x \rightarrow y) \vee (y \rightarrow x) = \mathbf{1} \tag{29}$$

*as well as the* divisibility *condition*

$$x * (x \rightarrow y) = x \wedge y\,. \tag{30}$$

The axiomatization of Hájek [22] for the *basic t-norm logic* BL (in [16] denoted BTL), i.e. for the class of all well-formed formulas which are valid in all BL-algebras, is given in a language $\mathcal{L}_T$ which has as basic vocabulary the connectives $\rightarrow, \&$ and the truth degree constant $\overline{0}$, taken in each BL-algebra $\langle L, \cap, \cup, *, \rightarrowtail, 0, 1\rangle$ as the operations $\rightarrowtail, *$ and the element 0. Then this t-norm based logic has as axiom system $\mathsf{Ax}_{\mathsf{BL}}$ the following schemata:

($\mathsf{Ax}_{\mathsf{BL}}1$) $(\varphi \rightarrow \psi) \rightarrow ((\psi \rightarrow \chi) \rightarrow (\varphi \rightarrow \chi))\,,$
($\mathsf{Ax}_{\mathsf{BL}}2$) $\varphi \,\&\, \psi \rightarrow \varphi\,,$
($\mathsf{Ax}_{\mathsf{BL}}3$) $\varphi \,\&\, \psi \rightarrow \psi \,\&\, \varphi\,,$
($\mathsf{Ax}_{\mathsf{BL}}4$) $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow (\varphi \,\&\, \psi \rightarrow \chi)\,,$
($\mathsf{Ax}_{\mathsf{BL}}5$) $(\varphi \,\&\, \psi \rightarrow \chi) \rightarrow (\varphi \rightarrow (\psi \rightarrow \chi))\,,$
($\mathsf{Ax}_{\mathsf{BL}}6$) $\varphi \,\&\, (\varphi \rightarrow \psi) \rightarrow \psi \,\&\, (\psi \rightarrow \varphi)\,,$
($\mathsf{Ax}_{\mathsf{BL}}7$) $((\varphi \rightarrow \psi) \rightarrow \chi) \rightarrow (((\psi \rightarrow \varphi) \rightarrow \chi) \rightarrow \chi)\,,$
($\mathsf{Ax}_{\mathsf{BL}}8$) $\overline{0} \rightarrow \varphi\,,$

and has as its (only) inference rule the rule of detachment, or: modus ponens (w.r.t. the implication connective →).

The logical calculus which is constituted by this axiom system and its inference rule, and which has the standard notion of derivation, shall be denoted by $\mathbb{K}_{\mathsf{BL}}$ or just by BL. (Similarly in other cases.)

Starting from the primitive connectives $\to, \&$ and the truth degree constant $\overline{0}$, the language $\mathcal{L}_T$ of $\mathsf{BL}$ is extended by definitions of additional connectives $\wedge, \vee, \neg$:

$$\begin{aligned} \varphi \wedge \psi &=_{\mathrm{df}} \varphi \,\&\, (\varphi \to \psi)\,, && (31)\\ \varphi \vee \psi &=_{\mathrm{df}} ((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi)\,, && (32)\\ \neg\varphi &=_{\mathrm{df}} \varphi \to \overline{0}\,. && (33) \end{aligned}$$

These additional connectives $\wedge, \vee$ just have the lattice operations $\cap, \cup$ as their truth degree functions.

It is a routine matter, but a bit tedious, to check that this logical calculus $\mathbb{K}_{\mathsf{BL}}$, usually called the axiomatic system $\mathsf{BL}$, is sound, i.e. derives only such formulas which are valid in all BL-algebras. A proof is given in [22], together with a proof of a corresponding completeness theorem.

**Corollary 1.** *The Lindenbaum algebra of the axiomatic system* $\mathsf{BL}$ *is a BL-algebra.*

**Theorem 2 (General Completeness).** *A formula $\varphi$ of the language $\mathcal{L}_T$ is derivable within the axiomatic system* $\mathsf{BL}$ *iff $\varphi$ is valid in all BL-algebras.*

The proof method yields that each BL-algebra is (isomorphic to) a subdirect product of linearly ordered BL-algebras, i.e. of BL-chains. Thus it allows a nice modification of the previous result.

**Corollary 2 (General Completeness; Version 2).** *A formula $\varphi$ of $\mathcal{L}_T$ is derivable within the axiomatic system* $\mathsf{BL}$ *iff $\varphi$ is valid in all BL-chains.*

But even more is provable and leads back to the starting point of the whole approach: the logical calculus $\mathbb{K}_{\mathsf{BL}}$ characterizes just those formulas which hold true w.r.t. all divisible t-norm algebras. This was proved in [4].

**Theorem 3 (Standard Completeness).** *The class of all formula which are provable in the system* $\mathsf{BL}$ *coincides with the class of all formulas which are logically valid in all t-norm algebras with a continuous t-norm.*

And another generalization of Theorem 2 deserves to be mentioned. To state it, let us call *schematic extension* of $\mathsf{BL}$ every extension which consists in an addition of finitely many axiom schemata to the axiom schemata of $\mathsf{BL}$. And let us denote such an extension by $\mathsf{BL}(\mathcal{C})$. And call BL($\mathcal{C}$)-algebra each BL-algebra $\mathbf{A}$ which makes $\mathbf{A}$-valid all formulas of $\mathcal{C}$.

Then one can prove, as done in [22], an even more general completeness result.

**Theorem 4 (Extended General Completeness).** *For each finite set $\mathcal{C}$ of axiom schemata and any formula $\varphi$ of $\mathcal{L}_T$ there are equivalent:*

*(i) $\varphi$ is derivable within* $\mathsf{BL}(\mathcal{C})$*;*
*(ii) $\varphi$ is valid in all* $\mathsf{BL}(\mathcal{C})$*-algebras;*
*(iii) $\varphi$ is valid in all* $\mathsf{BL}(\mathcal{C})$*-chains.*

The extension of these considerations to the first-order case is also given in [22], but shall not be discussed here.

But the algebraic machinery allows even deeper insights. After some particular results e.g. in [24, 25], the study of such subvarieties of the variety of all BL-algebras which are generated by single t-norm algebras of the form $\langle [0,1], \wedge, \vee, \otimes, \rightarrowtail, 0, 1\rangle$ with a continuous t-norm $\otimes$ led to (finite) axiomatizations of those t-norm based logics which have a standard semantics determined just by this continuous t-norm algebra. These results have been presented in [10].

### 3.6 The Logic of Left Continuous t-Norms

The guess of Esteva/Godo [9] has been that one should arrive at the logic of left continuous t-norms if one starts from the logic of continuous t-norms and deletes the continuity condition, i.e. the divisibility condition (27).

The algebraic approach needs only a small modification: in the Definition 3 of BL-algebras one has simply to delete the divisibility condition (30). The resulting algebraic structures have been called *MTL-algebras*. They again form a variety.

Following this idea, one has to modify the previous axiom system in a suitable way. And one has to delete the definition (31) of the connective $\wedge$, because this definition (together with suitable axioms) essentially codes the divisibility condition. The definition (32) of the connective $\vee$ remains unchanged.

As a result one now considers a new system $\mathsf{MTL}$ of mathematical fuzzy logic, known as *monoidal t-norm logic*, characterized semantically by the class of all MTL-algebras. It is connected with the axiom system

$(\mathsf{Ax}_{\mathsf{MTL}}1)$ $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}2)$ $\varphi \,\&\, \psi \to \varphi\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}3)$ $\varphi \,\&\, \psi \to \psi \,\&\, \varphi\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}4)$ $(\varphi \to (\psi \to \chi)) \to (\varphi \,\&\, \psi \to \chi)\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}5)$ $(\varphi \,\&\, \psi \to \chi) \to (\varphi \to (\psi \to \chi))\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}6)$ $\varphi \wedge \psi \to \varphi\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}7)$ $\varphi \wedge \psi \to \psi \wedge \varphi\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}8)$ $\varphi \,\&\, (\varphi \to \psi) \to \varphi \wedge \psi\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}9)$ $\overline{0} \to \varphi\,,$
$(\mathsf{Ax}_{\mathsf{MTL}}10)$ $((\varphi \to \psi) \to \chi) \to (((\psi \to \varphi) \to \chi) \to \chi)\,,$

together with the rule of detachment (w.r.t. the implication connective $\to$) as (the only) inference rule.

It is a routine matter, but again tedious, to check that this logical calculus $\mathbb{K}_{\mathsf{MTL}}$ is sound, i.e. derives only such formulas which are valid in all MTL-algebras.

**Corollary 3.** *The Lindenbaum algebra of the logical calculus* $\mathbb{K}_{\mathsf{MTL}}$ *is an MTL-algebra.*

Proofs of this result and also of the following completeness theorem are given in [9].

**Theorem 5 (General Completeness).** *A formula $\varphi$ of the language $\mathcal{L}_T$ is derivable within the logical calculus $\mathbb{K}_{\mathsf{MTL}}$ iff $\varphi$ is valid in all MTL-algebras.*

Again the proof method yields that each MTL-algebra is (isomorphic to) a subdirect product of linearly ordered MTL-algebras, i.e. of MTL-chains.

**Corollary 4 (General Completeness; Version 2).** *A formula $\varphi$ of $\mathcal{L}_T$ is derivable within the axiomatic system* MTL *iff $\varphi$ is valid in all MTL-chains.*

And again, similar as for the BL-case, even more is provable: the logical calculus $\mathbb{K}_{\mathsf{MTL}}$ characterizes just these formulas which hold true w.r.t. all those t-norm based logics which are determined by a left continuous t-norm. A proof is given in [27].

**Theorem 6 (Standard Completeness).** *The class of all formulas which are provable in the logical calculus $\mathbb{K}_{\mathsf{MTL}}$ coincides with the class of all formulas which are logically valid in all t-norm algebras with a left continuous t-norm.*

This result again means, as the similar one for the logic of continuous t-norms, that the variety of all MTL-algebras is the smallest variety which contains all t-norm algebras with a left continuous t-norm.

Because of the fact that the BL-algebras are the divisible MTL-algebras, one gets another adequate axiomatization of the basic t-norm logic BL if one extends the axiom system $\mathbb{K}_{\mathsf{MTL}}$ with the additional axiom schema

$$\varphi \wedge \psi \rightarrow \varphi \& (\varphi \rightarrow \psi)\,. \tag{34}$$

The simplest way to prove that this implication is sufficient is to show that the inequality $x * (x \rightarrowtail y) \leq x \cap y$, which corresponds to the converse implication, holds true in each MTL-algebra. Similar remarks apply to further extensions of MTL we are going to mention.

Also for MTL an extended completeness theorem similar to Theorem 4 remains true.

**Theorem 7 (Extended General Completeness).** *For each finite set $\mathcal{C}$ of axiom schemata and any formula $\varphi$ of $\mathcal{L}_T$ the following are equivalent:*

*(i) $\varphi$ is derivable within the logical calculus $\mathbb{K}_{\mathsf{MTL}} + \mathcal{C}$;*
*(ii) $\varphi$ is valid in all MTL($\mathcal{C}$)-algebras;*
*(iii) $\varphi$ is valid in all MTL($\mathcal{C}$)-chains.*

Again the extension to the first-order case is similar to the treatment in [22] for BL and shall not be discussed here.

The core point is that the formal language has to use predicate symbols to introduce atomic formulas, and that the logical apparatus has to be extended by quantifiers: and these are usually a generalization $\forall$ and a particularization $\exists$. As semantic interpretations of these quantifiers one uses in case of $\forall$ the infimum of the set of truth degrees of all the instances, and in case of $\exists$ the corresponding supremum. The semantic models $(\mathbf{M}, L)$ for this first-order language are determined by a nonempty universe $M$, a truth degree lattice $L$, and for each $n$-ary predicate symbol $P$ some $n$-ary $L$-valued fuzzy relation in $M$.

This forces either to assume that the truth degree lattices $L$ are complete lattices, or weaker and more preferably that the model $(\mathbf{M}, L)$ has to be a *safe* one, i.e. one for which all those subsets of $L$ have suprema and infima which may occur as truth degree sets of instances of quantified formulas of the language.

## 4 Linguistic Control and Fuzzy Relational Equations

### 4.1 The Standard Paradigm

The standard paradigm of rule based fuzzy control is that one supposes to have given, in a granular way, an incomplete and fuzzy description of a control function $\Phi$ from an input space $\mathbb{X}$ to an output space $\mathbb{Y}$, realized by a finite family

$$\mathcal{D} = (\langle A_i, B_i \rangle)_{1 \leq i \leq n} \tag{35}$$

of (fuzzy) input-output data pairs. These granular data are supposed to characterize this function $\Phi$ sufficiently well.

In the usual approaches such a family of input-output data pairs is provided by a finite list

$$\text{IF } \alpha \text{ is } A_i, \text{ THEN } \beta \text{ is } B_i, \quad i = 1, \ldots, n\,, \tag{36}$$

of linguistic control rules, also called fuzzy IF-THEN rules, describing some control procedure with input variable $\alpha$ and output variable $\beta$.

Mainly in engineering papers one often considers also the case of different input variables $\alpha_1, \ldots, \alpha_m$. In this case the linguistic control rules become the form

$$\text{IF } \alpha_1 \text{ IS } A_i^1, \text{ AND } \ldots \text{ AND } \alpha_m \text{ IS } A_i^m, \quad \text{THEN } \beta \text{ IS } B_i, \quad i = 1, \ldots, n\,. \tag{37}$$

But from a mathematical point of view such rules are subsumed among the former ones and cover only a restricted class of cases. To see this one simply has to allow as the input universe for $\alpha$ the cartesian product $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_m$ of the input universes $\mathbb{X}_i$ of $\alpha_i$, $i = 1, \ldots, m$. This yields for a given list $A_1, \ldots, A_m$ of input sets for the variables $\alpha_1, \ldots, \alpha_m$ the particular fuzzy input set $A = A_1 \times \cdots \times A_m$ for the combined variable $\alpha$. The above mentioned restriction comes from the fact that not all fuzzy subsets of $\mathbb{X}$ have this form of a fuzzy cartesian product of fuzzy subsets of the universes $\mathbb{X}_i$.

Let us assume for simplicity that all the input data $A_i$ are *normal*, i.e. that for each $i$ there is a point $x_0^i$ in the universe of discourse with $A_i(x_0^i) = 1$. Sometimes even weak normality would suffice, i.e. that the supremum over all the membership degrees of the $A_i$ equals one; but we do not indent to discuss this in detail.

The main mathematical problem of fuzzy control, besides the engineering problem to get a suitable list of linguistic control rules for the actual control problem, is therefore the interpolation problem to find a function $\Phi^* : \mathbb{F}(\mathbb{X}) \to \mathbb{F}(\mathbb{Y})$ which interpolates these data, i.e. which satisfies

$$\Phi^*(A_i) = B_i \quad \text{for each } i = 1, \ldots, n\,, \tag{38}$$

and which, in this way, gives a fuzzy representation for the control function $\Phi$.

Actually the standard approach is to look for *one* single function, more precisely: for some uniformly defined function, which should interpolate all these data, and which should be globally defined over the class $\mathbb{F}(\mathbb{X})$ of all fuzzy subsets of $\mathbb{X}$, or at least over a suitably chosen sufficiently large subclass of $\mathbb{F}(\mathbb{X})$.

Following Zadeh [36], this idea is formally realized by a fuzzy relation $R$, which connects fuzzy input information $A$ with fuzzy output information $B = A \circ R$ via the compositional rule of inference (3). Therefore, applying this idea to the linguistic control rules themselves, transforms these rules in a natural way into a system of fuzzy relation equations

$$A_i \circ R = B_i, \qquad \mathsf{for} \quad i = 1, \ldots, n\,. \tag{39}$$

The problem, to determine a fuzzy relation $R$ which realizes via (3) such a list (36) of linguistic control rules, becomes the problem to determine a solution of the corresponding system (39) of relation equations.

This problem proves to be a rather difficult one: it often happens that a given system (39) of relation equations is unsolvable. This is already the case in the more specific situation that the membership degrees belong to a Boolean algebra, as discussed (as a problem for Boolean matrices) e.g. in [30].

Nice solvability criteria are still largely unknown. Thus the investigation of the structure of the solution space for (39) was one of the problems discussed rather early. One essentially has that this space is an upper semilattice under the simple set union determined by the maximum of the membership degrees; cf. e.g. [6].

And this semilattice has, if it is nonempty, a universal upper bound

$$\widehat{R} = \bigcap_{i=1}^{n} \{(x,y) \parallel A_i(x) \to B_i(y)\} \tag{40}$$

as explained by the following result.

**Theorem 8.** *The system (39) of relation equations is solvable iff the fuzzy relation $\widehat{R}$ is a solution of it. And in the case of solvability, $\widehat{R}$ is always the largest solution of the system (39) of relation equations.*

This result was first stated in [35] for the particular case of the min-based Gödel implication $\to$ in (40), and generalized to the case of the residuated implications based upon arbitrary left continuous t-norms—and hence to the present situation—by this author in [13]; cf. also his [14].

Besides the reference to the CRI in this type of approach toward fuzzy control, the crucial point is to determine a fuzzy relation out of a list of linguistic control rules.

The fuzzy relation $\widehat{R}$ can be seen as a formalization of the idea that the list (36) of control rules has to be read as:

IF input is $A_1$ THEN output is $B_1$
AND
...
AND
IF input is $A_n$ THEN output is $B_n$ .

Having in mind such a formalization of the list (36) of control rules, there is immediately also another way how to read this list: substitute an OR for the AND to combine the single rules.

It is this understanding of the list of linguistic control rules as a (rough) description of a *fuzzy function* which characterizes the approach of Mamdani/Assilian [32]. Therefore they consider instead of $\widehat{R}$ the fuzzy relation

$$R_{\mathrm{MA}} = \bigcup_{i=1}^{n} (A_i \times B_i) ,$$

again combined with the compositional rule of inference.

### 4.2 Solutions and Pseudo-solutions of Fuzzy Relational Equations

Linguistic control rules serve as a tool for rough model building, and their translation into fuzzy relational equations does so too. So it may not really be necessary to solve systems of fuzzy relational equations (39) in the standard sense, but some kind of "approximation" of solutions may do as well.

There is, however, a a quite fundamental problem to understand this remark: an approximation is always an approximation of something. So an approximate solution of a system (39) should (normally) be an approximation of a true solution of this system. But what, if the system is not solvable at all?

A way out is offered by another type of approximation: by an approximation of the "behavior" of the functional which is given by the left hand sides of the equations from (39). In terms of the linguistic control rules which constituted (39) this means that one looks for an approximate realization of their behavior.

This way of doing, originating from [32] and the fuzzy relation $R_{\mathrm{MA}}$ introduced there, can also be seen as to "fake" something similar to a solution – and to work with it like a solution.

Such "faked" solutions shall be coined *pseudo-solutions*, following [21]. The best known pseudo-solutions for a system (39) are $\widehat{R}$ and $R_{\mathrm{MA}}$.

From Theorem 8 it is clear that $\widehat{R}$ is a pseudo-solution of a system (39) just in the case that the system (39) is not solvable. So one is immediately lead to the

**Problem:** Under which conditions is the pseudo-solution $R_{\mathrm{MA}}$ really a solution of the corresponding system (39) of relation equations.

This problem is discussed in [28]. And one of the main results is the next theorem.

**Theorem 9.** *Let all the input sets $A_i$ be normal. Then the fuzzy relation $R_{MA}$ is a solution of the corresponding system of fuzzy relation equations iff for all $i, j = 1, \dots, n$ one has*

$$\models \quad \exists x(A_i(x) \;\&\; A_j(x)) \to B_i \equiv^* B_j \,. \tag{41}$$

This MA-solvability criterion (41) is a kind of functionality of the list of linguistic control rules, at least in the case of the presence of an involutive negation: because in such a case one has

$$\models \quad \exists x(A_i(x) \;\&\; A_j(x)) \;\leftrightarrow\; A_i \cap_{\boldsymbol{t}} A_j \not\equiv^* \emptyset \,,$$

and thus condition (41) becomes

$$\models \quad A_i \cap_{\boldsymbol{t}} A_j \not\equiv^* \emptyset \;\to\; B_i \equiv^* B_j \,. \tag{42}$$

And this can be understood as a fuzzification of the idea *"if $A_i$ and $A_j$ coincide to some degree, than also $B_i$ and $B_j$ should coincide to a certain degree"*.

Of course, this fuzzification is neither obvious nor completely natural, because it translates "degree of coincidence" in two different ways.

This leads back to the well known result, explained e.g. in [14], that the system of relation equations is solvable in the case that all the input fuzzy sets $A_i$ are pairwise $\boldsymbol{t}$-disjoint:

$$A_i \cap_{\boldsymbol{t}} A_j = \emptyset \quad \text{for all } i \neq j.$$

Here the reader should have in mind that this $\boldsymbol{t}$-disjointness is, in general, weaker than the standard min-disjointness: it does not force the disjointness of the supports of the fuzzy sets $A_i, A_j$, and even allows a height $\mathsf{hgt}(A_i \cap A_j) = 0.5$ for the case that $\boldsymbol{t}$ is the Łukasiewicz t-norm.

However, it may happen that the system of relation equations is solvable, i.e. has $\widehat{R}$ as a solution, without having the fuzzy relation $R_{\mathrm{MA}}$ as a solution.

An example is given in [21].

Therefore condition (41) is only a sufficient one for the solvability of the system (39) of relation equations.

Hence one has as a new problem to give additional assumptions, besides the solvability of the system (39) of relation equations, which are sufficient to guarantee that $R_{\mathrm{MA}}$ is a solution of (39).

As in [14] and already in [12], we subdivide the problem whether a fuzzy relation $R$ is a solution of the system of relation equations into two cases: (i) whether one has satisfied the *subset property* w.r.t. a system (39), i.e. whether one has

$$A_i \circ R \subset B_i, \quad \text{for} \quad i = 1, \dots, n \,, \tag{43}$$

and (ii) whether one has the *superset property* w.r.t. (39), i.e. whether one has one has

$$A_i \circ R \supset B_i, \quad \text{for} \quad i = 1, \dots, n \,. \tag{44}$$

The core result is the following theorem.

**Theorem 10.** *If the input set $A_k$ is normal then* $A_k \circ \widehat{R} \subset B_k \subset A_k \circ R_{\mathrm{MA}}$.

So we know that with normal input sets the fuzzy outputs $A_i \circ \widehat{R}$ always are fuzzy subsets of $A_i \circ R_{\mathrm{MA}}$.

Furthermore we immediately have the following global result.

**Proposition 2.** *If all the input sets $A_i$ of the system of relation equations are normal and if one also has $R_{\mathrm{MA}} \subset \widehat{R}$, then the system of relation equations is solvable, and $R_{\mathrm{MA}}$ is a solution.*

Hence the pseudo-solutions $R_{\mathrm{MA}}$ and $\widehat{R}$ are upper and lower approximations for the realizations of the linguistic control rules.

Now one may equally well look for new pseudo-solutions, e.g. by some *iteration* of these pseudo-solutions in the way, that for the *next iteration step* in such an iteration process the system of relation equations is changed such that its (new) output sets become the *real output* of the *former* iteration step. This has been done in [21].

To formulate the dependence of the pseudo-solutions $R_{\mathrm{MA}}$ and $\widehat{R}$ from the input and output data, we denote the "original" pseudo-solutions with the input-output data $(A_i, B_i)$ in another way and write

$$R_{\mathrm{MA}}[B_k] \ \text{ for } \ R_{\mathrm{MA}} \,, \qquad \widehat{R}[B_k] \ \text{ for } \ \widehat{R} \,.$$

**Theorem 11.** *One has always*

$$A_i \circ \widehat{R}[B_k] \subset A_i \circ R_{\mathrm{MA}}[A_k \circ \widehat{R}[B_k]] \subset A_i \circ R_{\mathrm{MA}}[B_k] \,.$$

Thus the iterated relation $R_{\mathrm{MA}}[A_k \circ \widehat{R}]$ is a pseudo-solution which somehow better approximates the intended behavior of the linguistic control rules as each one of $R_{\mathrm{MA}}$ and $\widehat{R}$. For details cf. again [21].

### 4.3 Invoking More Formal Logic

The languages of the first-order versions of any one of the standard fuzzy logics discussed in Section 3 can be used to formalize the main ideas behind the use of linguistic control rules in fuzzy control matters. This offers, besides the above presented reference to fuzzy set and fuzzy relation matters, a second way for a mathematical analysis of this rough modeling strategy.

We shall use here the logic $\mathsf{BL}\forall$, i.e. the first-order version of the logic $\mathsf{BL}$ of continuous t-norms as the basic reference logic for this analysis. And we follow in the presentation of the material closely the paper [33].

A formula of this logic, with $n$ free variables, describes w.r.t. each $\mathsf{BL}\forall$-model $(\mathbf{M}, L)$ some $n$-ary $L$-fuzzy relation in the universe $M = |\mathbf{M}|$. This is the obvious specification of the approach of Section 2 to describe fuzzy sets by formulas of a suitable language.

A BL-theory $\mathcal{T}$ is just a crisp, i.e. classical set of formulas of the language of $\mathsf{BL}\forall$. The notation $\mathcal{T} \vdash A$ means that the formula $A$ is provable in a theory $\mathcal{T}$.

In what follows, we will use the letters $A, B, \ldots$ formulas as well as for $L$-fuzzy sets in some $\mathsf{BL}\forall$-model $(\mathbf{M}, L)$.

This yields that e.g. the result (3) of a – now t-norm based and not only min-based – CRI-application to a fuzzy set $A$ and a fuzzy relation $R$ is described by the formula

$$(\exists x)(A(x) \mathbin{\&} R(x, y))\,. \tag{45}$$

Furthermore, if $\mathcal{T}$ is a BL-theory and $\mathcal{T} \vdash (\exists x)A(x)$ then this means that the fuzzy set described by $A$ is normal.

Let us additionally assume that $\mathcal{T}$ is a consistent $\mathsf{BL}\forall$-theory which formalizes some additional assumptions which may be made for the considerations of a fuzzy control problem, e.g. that the input fuzzy sets may be normal ones.

On this basis, the results in fuzzy relation equations can be succinctly and transparently formulated in the syntax of $\mathsf{BL}\forall$. So let us start from a system (36) of linguistic control rules with input fuzzy sets $A_i$ over some $m$-dimensional universe $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_m$.

With our notational convention this means that we have given formulas $A_i(x_1, \ldots, x_m)$ with free variables $x_1, \ldots, x_m$ which describe these input sets of the single control rules. For simplicity, let us write $\overrightarrow{x} = (x_1, \ldots, x_m)$ for this list of (free) variables.

The problem which corresponds to the solvability problem of a system (39) of relation equations now is to find a formula $R(\overrightarrow{x}, y)$ with $m + 1$ free variables such that

$$\mathcal{T} \vdash (\exists x_1) \cdots (\exists x_n)(A_i(\overrightarrow{x}) \mathbin{\&} R(\overrightarrow{x}, y)) \leftrightarrow B_i(y) \tag{46}$$

holds for every $i = 1, \ldots, n$. If such a formula $R$ exists then we say that the *system of fuzzy relation equations* (46) is *solvable in* $\mathcal{T}$ and $R(\overrightarrow{x}, y)$ is its solution.

**Lemma 1.** *The following is* $\mathsf{BL}\forall$*-provable:*

$$\mathcal{T} \vdash (\exists x_1) \cdots (\exists x_m) \left( A_i(\overrightarrow{x}) \mathbin{\&} \bigwedge_{j=1}^{n} (A_j(\overrightarrow{x}) \to B_j(y)) \right) \to B_i(y).$$

The following theorem presents in a purely syntactical way the well known fundamental result on the solution of the fuzzy relation equations.

**Theorem 12.** *The system of fuzzy relation equations (46) is solvable in* $\mathcal{T}$ *iff*

$$\mathcal{T} \vdash (\exists x_1) \cdots (\exists x_m) \left( A_i(\overrightarrow{x}) \mathbin{\&} \bigwedge_{j=1}^{n} (A_j(\overrightarrow{x}) \to B_j(y)) \right) \leftrightarrow B_i(y) \tag{47}$$

*holds for all* $i = 1, \ldots, n$.

**Corollary 5.** *If (46) holds for some* $R(\overrightarrow{x}, y)$ *then*

$$\mathcal{T} \vdash R(\overrightarrow{x}, y) \to \bigwedge_{j=1}^{n} (A_j(\overrightarrow{x}) \to B_j(y))\,.$$

This means, in the present terminology, that the solution (47) is maximal.

To simplify the formulas to come let us assume for the rest of this subsection that $m = 1$. This corresponds completely to the former subsumption of the case (37) of a system of linguistic control rules with $m$ input variables under the case (36) with only one input variable.

We will work with the following two special kinds of formulas. The first one has the form of a conjunction of implications from Theorem 12

$$R^{Imp}(x, y) =_{\mathrm{def}} \bigwedge_{j=1}^{m} (A_j(x) \to B_j(y)))\,. \tag{48}$$

The second one, the so called Mamdani-Assilian formula, shall be

$$R^{MA}(x, y) = \bigvee_{i=1}^{m} (A_i(x) \;\&\; B_i(y))\,. \tag{49}$$

**Lemma 2.** *Let $\mathcal{T} \vdash (\exists x)A_i$ for all $i = 1, \ldots, n$. Then*

$$\mathcal{T} \vdash B_i(y) \to (\exists x)(A_i(x) \;\&\; \bigvee_{j=1}^{m} (A_j(x) \;\&\; B_j(y))). \tag{50}$$

Joining Lemmas 1 and 2, we get get following theorem, which says that our conjunction (48) of implications gives the lower and the Mamdani-Assilian formula (49) the upper bound for the solutions of the system (46).

**Theorem 13.** *Let $\mathcal{T} \vdash (\exists x)A_i$ for $i = 1, \ldots, n$. Then the following is provable:*

$$\begin{aligned} &\mathcal{T} \vdash (\exists x)(A_i(x) \;\&\; R^{Impl}(x, y)) \to B_i(y)\,, \\ &\mathcal{T} \vdash B_i(y) \to (\exists x)(A_i(x) \;\&\; R^{MA}) \end{aligned}$$

*for each $i = 1, \ldots, n$.*

The following theorem has a semantical counterpart proved with some more restrictive assumptions by Perfilieva/Tonis [34].

**Theorem 14.** *Let $A(x)$ be an arbitrary formula. The following is provable for each $i = 1, \ldots, n$:*

$$\begin{aligned} \mathcal{T} \vdash (A(x) \leftrightarrow A_i(x)) \;\&\; (B_i(y) \to (\exists x)(A_i(x) \;\&\; R^{Impl}(x, y))) \to \\ ((\exists x)(A(x) \;\&\; R^{Impl}(x, y)) \leftrightarrow B_i(y))\,. \end{aligned} \tag{51}$$

This theorem suggests that the formula

$$\xi(y) =_{\mathrm{def}} (B_i(y) \to (\exists x)(A_i(x) \;\&\; R^{Impl}(x, y)))$$

can be the basis of a *solvability sign* similar to a more general solvability index discussed by this author e.g. in [14, 15] and defined as $\xi(y) =_{\mathrm{def}} \bigwedge_{i=1}^{n} \xi_i(y)$.

Then from Theorem 14 it follows:

**Corollary 6**

$$\mathcal{T} \vdash \xi(y) \rightarrow ((\exists x)(A_i(x) \,\&\, R^{Impl}(x,y)) \leftrightarrow B_i(y))$$

*for each* $i = 1, \ldots, n$.

It follows from this corollary that in a model of $\mathcal{T}$, the solvability sign can be interpreted as a degree, in which the system (46) is solvable in $\mathcal{T}$.

**Theorem 15.** *The system (46) is solvable in* $\mathcal{T}$ *iff*

$$\mathcal{T} \vdash (\forall y)\xi(y).$$

**Theorem 16.** *If the system (46) is solvable in* $\mathcal{T}$ *then*

$$\mathcal{T} \vdash (\forall x)(A_i(x) \leftrightarrow A_j(x)) \rightarrow (\forall y)(B_i(y) \leftrightarrow B_j(y)) \tag{52}$$

*for all* $i, j = 1, \ldots, n$.

Klawonn [28] gave a necessary and sufficient condition that the Mamdani-Assilian fuzzy relation is a solution of a system of fuzzy relation equations. The formalized version of this result can again be $\mathsf{BL}\forall$-proved.

**Theorem 17.** *If* $T \vdash (\exists x)A_i(x)$, $i = 1, \ldots, m$. *Then* $R^{MA}(x,y)$ *is a solution of (46) iff*

$$\mathcal{T} \vdash (\exists x)(A_i(x) \,\&\, A_j(x)) \rightarrow (\forall y)(B_i(y) \leftrightarrow B_j(y)), \qquad i = 1, \ldots, n. \tag{53}$$

The following theorem is a corollary of the previous results.

**Corollary 7.** $(i)$ *If there is an index* $k$ *such that*

$$\mathcal{T} \not\vdash (\exists y)B_k(y) \rightarrow (\exists x)A_k(x)\,, \tag{54}$$

*then the system (46) is not solvable.*

$(ii)$ *Assume*

$$\mathcal{T} \not\vdash (\exists x)(\exists y)\bigvee_{j=1}^{n}(A_i(x) \,\&\, B_i(y)) \rightarrow (\exists x)(\exists y)R(x,y)\,.$$

*Then* $R(x,y)$ *is not a solution of (46).*

## 5 Approximation and Interpolation

The standard mathematical understanding of *approximation* is that by an approximation process some mathematical object $A$, e.g. some function, is approximated, i.e. determined within some (usually previously unspecified) error bounds.

Additionally one assumes that the approximating object $B$ for $A$ is of some predetermined, usually "simpler" kind, e.g. a polynomial function.

So one may approximate some transcendental function, e.g. the trajectory of some non-linear process, by a piecewise linear function, or by a polynomial function of some bounded degree. Similarly one approximates e.g. in the Runge-Kutta methods the solution of a differential equation by a piecewise linear function, or one uses splines to approximate a difficult surface in 3-space by planar pieces.

The standard mathematical understanding of *interpolation* is that a function $f$ is only partially given by its values *at some points* of the domain of the function, the interpolation nodes.

The problem then is to determine "the" values of $f$ for all the other points of the domain (usually) *between* the interpolation nodes – sometimes also outside these interpolation nodes (extrapolation).

And this is usually done in such a way that one considers groups of neighboring interpolation nodes which uniquely determine an interpolating function of some predetermined type within their convex hull (or something like): a function which has the interpolation nodes of the actual group as argument-value pairs – and which in this sense *locally* approximates the function $f$.

In the *standard fuzzy control approach* the input-output data pairs of the linguistic control rules just provide interpolation nodes.

However, what is lacking – at least up to now – that is the idea of a *local approximation* of the intended crisp control function by some fuzzy function. Instead, in the standard contexts one always asks for something like a *global* interpolation, i.e. one is interested to interpolate *all* nodes by *only one* interpolation function.

To get a local approximation of the intended crisp control function $\Phi$, one needs some notion of "nearness" or of "neighboring" for fuzzy data granules. Such a notion is lacking in general.

For the particular case of a linearly ordered input universe $\mathbb{X}$, and the additional assumption that the fuzzy input data are unimodal, one gets in a natural way from this crisp background a notion of neighboring interpolation nodes: fuzzy nodes are neighboring if their kernel points are.

In general, however, it seems most appropriate to suppose that one may be able to infer from the control problem a—perhaps itself fuzzy—partitioning of the whole input space (or similarly of the output space). Then one will be in a position to split in a natural way the data set (35), or correspondingly the list (36) of control rules, into different groups—and to consider the localized interpolation problems separately for these groups.

This offers obviously better chances for finding interpolating functions, particularly for getting solvable systems of fuzzy relation equations. However, one has to be aware that one should additionally take care that the different local interpolation functions fit together somehow smoothly—again an open problem that needs a separate discussion. It is a problem that is more complicated for fuzzy interpolation than for the crisp counterpart because the fuzzy interpolating functions may realize the fuzzy interpolation nodes only approximately.

However, one may start from ideas like these to speculate about fuzzy versions of the standard *spline interpolation* methodology.

In the context of fuzzy control the control function $\Phi$, which has to be determined, is described only roughly, i.e. given only by its behavior in some (fuzzy) points of the state space.

The standard way to roughly describe the control function is to give a list (36) of linguistic control rules connecting fuzzy subsets $A_i$ of the input space $\mathbb{X}$ with fuzzy subsets $B_i$ of the output space $\mathbb{Y}$ indicating that one likes to have

$$\Phi^*(A_i) = B_i\,, \quad i = 1, \dots, n \tag{55}$$

for a suitable "fuzzified" version $\Phi^* : \mathbb{F}(\mathbb{X}) \to \mathbb{F}(\mathbb{Y})$ of the control function $\Phi : \mathbb{X} \to \mathbb{Y}$.

The additional approximation idea of the CRI is to approximate $\Phi^*$ by a fuzzy function $\Psi^* : \mathbb{F}(\mathbb{X}) \to \mathbb{F}(\mathbb{Y})$ determined for all $A \in \mathbb{F}(\mathbb{X})$ by

$$\Psi^*(A) = A \circ R \tag{56}$$

which refers to some suitable fuzzy relation $R \in \mathbb{F}(\mathbb{X} \times \mathbb{Y})$, and understands $\circ$ as sup-t-composition.

Formally thus the equations (55) become transformed into some well known system (39) of relation equations

$$A_i \circ R = B_i, \qquad i = 1, \dots, n$$

to be solved for the unknown fuzzy relation $R$.

This approximation idea fits well with the fact that one often is satisfied with pseudo-solutions of (39), and particularly with the MA-pseudo-solution $R_{\mathrm{MA}}$, or the S-pseudo-solution $\widehat{R}$. Both of them determine approximations $\Psi^*$ to the (fuzzified) control function $\Phi^*$.

What remains open in this discussion up to now are quality considerations for such approximations via pseudo-solutions of systems (39) of fuzzy relational equations. This is a topic which has been discussed only scarcely. Qualitative approaches referring to the ideas of upper and lower approximations w.r.t. fuzzy inclusion, as considered here in Subsection 4.2, have been considered in [17] and extensively reported in [20]. The interested reader may consult these sources. More quantitative approaches, based e.g. upon the consideration of metrics or pseudo-metrics in spaces of fuzzy sets, are missing up to now – as far as this author is aware of.

## 6 Aggregation Operations and Fuzzy Control Strategies

There is the well known distinction between FATI and FITA strategies to evaluate systems of linguistic control rules w.r.t. arbitrary fuzzy inputs from $\mathbb{F}(\mathbb{X})$.

The core idea of a FITA strategy is that it is a strategy which **F**irst **I**nfers (by reference to the single rules) and **T**hen **A**ggregates starting from the actual input information $A$. Contrary to that, a FATI strategy is a strategy which **F**irst **A**ggregates (the information in all the rules into one fuzzy relation) and **T**hen **I**nfers starting from the actual input information $A$.

Both these strategies use the set theoretic union as their aggregation operator. Furthermore, both of them refer to the CRI as their core tool of inference.

In general, however, the interpolation operators may depend more generally upon some inference operator(s) as well as upon some aggregation operator.

By an *inference operator* we mean here simply a mapping from the fuzzy subsets of the input space to the fuzzy subsets of the output space.[3]

And an *aggregation operator* $\mathbf{A}$, as explained e.g. in [2, 7], is a family $(f^n)_{n\in\mathbb{N}}$ of ("aggregation") operations, each $f^n$ an $n$-ary one, over some partially ordered set $\mathbf{M}$, with ordering $\leqslant$, with a bottom element $\mathbf{0}$ and a top element $\mathbf{1}$, such that each operation $f^n$ is non-decreasing, maps the bottom to the bottom: $f^n(\mathbf{0},\ldots,\mathbf{0})=\mathbf{0}$, and the top to the top: $f^n(\mathbf{1},\ldots,\mathbf{1})=\mathbf{1}$.

Such an aggregation operator $\mathbf{A}=(f^n)_{n\in\mathbb{N}}$ is a *commutative* one iff each operation $f^n$ is commutative. And $\mathbf{A}$ is an *associative* aggregation operator iff

$$f^n(a_1,\ldots,a_n)=f^r(f^{k_1}(a_1,\ldots,a_{k_1}),\ldots,f^{k_r}(a_{m+1},\ldots,a_n))$$

for $n=\sum_{i=1}^{r}k_i$ and $m=\sum_{i=1}^{r-1}k_i$.

Our aggregation operators further on are supposed to be commutative as well as associative ones.[4]

As in [18], we now consider interpolation operators $\Phi$ of FITA-type and interpolation operators $\Psi$ of FATI-type which have the abstract forms

$$\Psi_{\mathcal{D}}(A)=\mathbf{A}(\theta_{\langle A_1,B_1\rangle}(A),\ldots,\theta_{\langle A_n,B_n\rangle}(A))\,, \tag{57}$$

$$\Xi_{\mathcal{D}}(A)=\widehat{\mathbf{A}}(\theta_{\langle A_1,B_1\rangle},\ldots,\theta_{\langle A_n,B_n\rangle})(A)\,. \tag{58}$$

Here we assume that each one of the "local" inference operators $\theta_i$ is determined by the single input-output pair $\langle A_i,B_i\rangle$. This restriction is sufficient for the present purpose.

In this Section we survey the main notions and results. The interested reader gets more details from [19] and also from [20].

### 6.1 Stability Conditions

If $\Theta_{\mathcal{D}}$ is a fuzzy inference operator of one of the types (57), (58), then the interpolation property one likes to have realized is that one has

$$\Theta_{\mathcal{D}}(A_i)=B_i \tag{59}$$

for all the data pairs $\langle A_i,B_i\rangle$. In the particular case that the operator $\Theta_{\mathcal{D}}$ is given by (3), this is just the problem to solve the system (59) of fuzzy relation equations.

[3] This terminology has its historical roots in the fuzzy control community. There is no relationship at all with the logical notion of inference intended and supposed here; but–of course–also not ruled out.

[4] It seems that this is a rather restrictive choice from a theoretical point of view. However, in all the usual cases these restrictions are satisfied.

**Definition 4.** *In the present generalized context let us call the property (59) the* $\mathcal{D}$-stability *of the fuzzy inference operator* $\Theta_{\mathcal{D}}$*.*

To find $\mathcal{D}$-stability conditions on this abstract level seems to be rather difficult in general. However, the restriction to fuzzy inference operators of FITA-type makes things easier.

It is necessary to have a closer look at the aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ involved in (57) which operates on $\mathbb{F}(\mathbb{Y})$, of course with the inclusion relation for fuzzy sets as partial ordering.

**Definition 5.** *Having* $B, C \in \mathbb{F}(\mathbb{Y})$ *we say that* $C$ *is* $\mathbf{A}$-negligible w.r.t. $B$ *iff* $f^2(B, C) = f^1(B)$ *holds true.*

The core idea here is that in any aggregation by $\mathbf{A}$ the presence of the fuzzy set $B$ among the aggregated fuzzy sets makes any presence of $C$ superfluous.

Hence one e.g. has that $C$ is $\bigcup$-negligible w.r.t. $B$ iff $C \subset B$; and $C$ is $\bigcap$-negligible w.r.t. $B$ iff $C \supset B$.

**Proposition 3.** *Consider a fuzzy inference operator* $\Psi_{\mathcal{D}}$ *of FITA-type (57). It is sufficient for the* $\mathcal{D}$*-stability of* $\Psi_{\mathcal{D}}$*, i.e. to have*

$$\Psi_{\mathcal{D}}(A_k) = B_k \qquad \textit{for all } k = 1, \ldots, n$$

*that one always has*

$$\theta_{\langle A_k, B_k \rangle}(A_k) = B_k$$

*and additionally that for each* $i \neq k$ *the fuzzy set*

$$\theta_{\langle A_k, B_k \rangle}(A_i) \quad \textit{is } \mathbf{A}\textit{-negligible w.r.t.} \quad \theta_{\langle A_k, B_k \rangle}(A_k)\,.$$

This result has two quite interesting specializations which themselves generalize well known results about fuzzy relation equations.

**Corollary 8.** *It is sufficient for the* $\mathcal{D}$*-stability of a fuzzy inference operator* $\Psi_{\mathcal{D}}$ *of FITA-type that one has*

$$\Psi_{\mathcal{D}}(A_i) = B_i \qquad \textit{for all } \ 1 \leq i \leq n$$

*and that always* $\theta_{\langle A_i, B_i \rangle}(A_j)$ *is* $\mathbf{A}$*-negligible w.r.t.* $\theta_{\langle A_i, B_i \rangle}(A_i)$*.*

To state the second one of these results, call an aggregation operator $\mathbf{A} = (f^n)_{n \in \mathbb{N}}$ *additive* iff always $b \leqslant f^2(b, c)$, and call it *idempotent* iff always $b = f^2(b, b)$.

**Corollary 9.** *It is sufficient for the* $\mathcal{D}$*-stability of a fuzzy inference operator* $\Psi_{\mathcal{D}}$ *of FITA-type, which is based upon an additive and idempotent aggregation operator, that one has*

$$\Psi_{\mathcal{D}}(A_i) = B_i \qquad \textit{for all } \ 1 \leq i \leq n$$

*and that always* $\theta_{\langle A_i, B_i \rangle}(A_j)$ *is the bottom element in the domain of the aggregation operator* $\mathbf{A}$*.*

Obviously this is a direct generalization of the fact that systems of fuzzy relation equations are solvable if their input data form a pairwise disjoint family (w.r.t. the corresponding t-norm based intersection $\cap$ and cartesian product $\times$) because in this case one has usually:

$$\begin{aligned}\theta_{\langle A_i,B_i\rangle}(A_j) &= A_j \circ (A_i \times B_i) = \{y \,\|\, \exists x(x \,\varepsilon\, A_j \,\&\, (x,y) \,\varepsilon\, A_i \times B_i)\} \\ &= \{y \,\|\, \exists x(x \,\varepsilon\, A_j \cap A_i \,\&\, y \,\varepsilon\, B_i)\}\,.\end{aligned}$$

To extend these considerations from inference operators (57) of the FITA type to those ones of the FATI type (58) let us consider the following notion.

**Definition 6.** *Suppose that $\widehat{\mathbf{A}}$ is an aggregation operator for inference operators, and that $\mathbf{A}$ is an aggregation operator for fuzzy sets. Then $(\widehat{\mathbf{A}}, \mathbf{A})$ is an* application distributive *pair of aggregation operators iff*

$$\widehat{\mathbf{A}}(\theta_1, \ldots, \theta_n)(X) = \mathbf{A}(\theta_1(X), \ldots, \theta_n(X)) \tag{60}$$

*holds true for arbitrary inference operators $\theta_1, \ldots, \theta_n$ and fuzzy sets $X$.*

Using this notion it is easy to see that one has on the left hand side of (60) a FATI type inference operator, and on the right hand side an associated FITA type inference operator. So one is able to give a reduction of the FATI case to the FITA case, assuming that such application distributive pairs of aggregation operators exist.

**Proposition 4.** *Suppose that $(\widehat{\mathbf{A}}, \mathbf{A})$ is an application distributive pair of aggregation operators. Then a fuzzy inference operator $\Xi_{\mathcal{D}}$ of FATI-type is $\mathcal{D}$-stable iff its associated fuzzy inference operator $\Psi_{\mathcal{D}}$ of FITA-type is $\mathcal{D}$-stable.*

The general approach of this Section can also be applied to the problem of $\mathcal{D}$-stability for a modified operator $\Theta^*_{\mathcal{D}}$ which is determined by the kind of iteration of $\Theta_{\mathcal{D}}$ which previously led to Theorem 11. To do this, let us consider the $\Theta_{\mathcal{D}}$-*modified* data set $\mathcal{D}^*$ given as

$$\mathcal{D}^* = (\langle A_i, \Theta_{\mathcal{D}}(A_i)\rangle)_{1\le i\le n}\,, \tag{61}$$

and define from it the modified fuzzy inference operator $\Theta^*_{\mathcal{D}}$ as

$$\Theta^*_{\mathcal{D}} = \Theta_{\mathcal{D}^*}\,. \tag{62}$$

For these modifications, the problem of stability reappears. Of course, the new situation here is only a particular case of the former. And it becomes a simpler one in the sense that the stability criteria now refer only to the input data $A_i$ of the data set $\mathcal{D} = (\langle A_i, B_i\rangle)_{1\le i\le n}$.

**Proposition 5.** *It is sufficient for the $\mathcal{D}^*$-stability of a fuzzy inference operator $\Psi^*_{\mathcal{D}}$ of FITA-type that one has*

$$\Psi^*_{\mathcal{D}}(A_i) = \Psi_{\mathcal{D}^*}(A_i) = \Psi_{\mathcal{D}}(A_i) \qquad \textit{for all}\;\; 1 \le i \le n \tag{63}$$

*and that always $\theta_{\langle A_i,\Psi_{\mathcal{D}}(A_i)\rangle}(A_j)$ is $\mathbf{A}$-negligible w.r.t. $\theta_{\langle A_i,\Psi_{\mathcal{D}}(A_i)\rangle}(A_i)$.*

Let us look separately at the conditions (63) and at the negligibility conditions.

**Corollary 10.** *The conditions* (*63*) *are always satisfied if the inference operator* $\Psi^*_{\mathcal{D}}$ *is determined by the standard output-modified system of relation equations* $A_i \circ R[A_k \circ R] = B_i$ *in the notation of* [*21*].

**Corollary 11.** *In the case that the aggregation operator is the set theoretic union, i.e.* $A = \bigcup$, *the conditions* (*63*) *together with the inclusion relationships*

$$\theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i)\rangle}(A_j) \subset \theta_{\langle A_i, \Psi_{\mathcal{D}}(A_i)\rangle}(A_i)$$

*are sufficient for the* $\mathcal{D}^*$*-stability of a fuzzy inference operator* $\Psi^*_{\mathcal{D}}$.

Again one is able to transfer this result to FATI-type fuzzy inference operators.

**Corollary 12.** *Suppose that* $(\widehat{\mathbf{A}}, \mathbf{A})$ *is an application distributive pair of aggregation operators. Then a fuzzy inference operator* $\Phi^*_{\mathcal{D}}$ *of FATI-type is* $\mathcal{D}^*$*-stable iff its associated fuzzy inference operator* $\Psi^*_{\mathcal{D}}$ *of FITA-type is* $\mathcal{D}^*$*-stable.*

### 6.2 Application Distributivity

Based upon the notion of application distributive pair of aggregation operators the property of $\mathcal{D}$-stability can be transferred back and forth between two inference operators of FATI-type and of FITA-type if they are based upon a pair of application distributive aggregation operators.

What has not been discussed previously was the existence and the uniqueness of such pairs. Here are some results concerning these problems.

The uniqueness problem has a simple solution.

**Proposition 6.** *If* $(\widehat{\mathbf{A}}, \mathbf{A})$ *is an application distributive pair of aggregation operators then* $\widehat{\mathbf{A}}$ *is uniquely determined by* $\mathbf{A}$, *and conversely also* $\mathbf{A}$ *is uniquely determined by* $\widehat{\mathbf{A}}$.

And for the existence problem we have a nice reduction to the two-argument case.

**Theorem 18.** *Suppose that* $\mathbf{A}$ *is a commutative and associative aggregation operator. For the case that there exists an aggregation operator* $\widehat{\mathbf{A}}$ *such that* $(\widehat{\mathbf{A}}, \mathbf{A})$ *form an application distributive pair of aggregation operators it is necessary and sufficient that there exists some operation* $G$ *for fuzzy inference operators satisfying*

$$\mathbf{A}(\theta_1(X), \theta_2(X)) = G(\theta_1, \theta_2)(X) \tag{64}$$

*for all fuzzy inference operators* $\theta_1, \theta_2$ *and all fuzzy sets* $X$.

However, there is an important restriction concerning the existence of such pairs of application distributive aggregation operators, at least for an interesting particular case.

**Definition 7.** *An aggregation operator* $\mathbf{A} = (f^n)_{n\in\mathbb{N}}$ *for fuzzy subsets of a common universe of discourse* $\mathbb{X}$ *is* pointwise defined *iff for each* $n \in \mathbb{N}$ *there exists a function* $g_n : [0,1]^n \to [0,1]$ *such that for all* $A_1, \dots, A_n \in \mathcal{F}$ *and all* $x \in \mathbf{X}$ *there hold*

$$f^n(A_1, \dots, A_n)(x) = g_n(A_1(x), \dots, A_n(x)) \,. \tag{65}$$

*And an aggregation operator* $\widehat{\mathbf{A}}$ *for inference operators is* pointwise defined *iff it can be reduced to a pointwise defined aggregation operator for fuzzy relations.*

The restrictive result, proved in [26], now reads as follows.

**Proposition 7.** *Among the commutative, associative, and pointwise defined aggregation operators is* $(\bigcup, \bigcup)$ *the only application distributive pair.*

### 6.3 Invoking a Defuzzification Strategy

In a lot of practical applications of the fuzzy control strategies which form the starting point for the previous general considerations, the fuzzy model—e.g. determined by a list (36) of linguistic IF-THEN-rules—is realized in the context of a further *defuzzification strategy*, which is nothing but a mapping $F : \mathbb{F}(\mathbb{Y}) \to \mathbb{Y}$ for fuzzy subsets of the output space $\mathbb{Y}$.

Having this in mind, it seems to be reasonable to consider the following modification of the $\mathcal{D}$-stability condition, which is a formalization of the idea to have "stability modulo defuzzification".

**Definition 8.** *A fuzzy inference operator* $\Theta_{\mathcal{D}}$ *is* $(F, \mathcal{D})$*-stable w.r.t. a defuzzification strategy* $F : \mathbb{F}(\mathbb{Y}) \to \mathbb{Y}$ *iff one has*

$$F(\Theta_{\mathcal{D}}(A_i)) = F(B_i) \tag{66}$$

*for all the data pairs* $\langle A_i, B_i\rangle$ *from* $\mathcal{D}$*.*

For the fuzzy modeling process which is manifested in the data set $\mathcal{D}$ this condition (66) is supposed to fit well with the control behavior one is interested to implement. If for some application this condition (66) seems to be unreasonable, this indicates that either the data set $\mathcal{D}$ or the chosen defuzzification strategy $F$ are unsuitable.

As a first, and rather restricted stability result for this modified situation, the following Proposition shall be mentioned.

**Proposition 8.** *Suppose that* $\Theta_{\mathcal{D}}$ *is a fuzzy inference operator of FITA-type (57), that the aggregation is union* $\mathbf{A} = \bigcup$ *as e.g. in the fuzzy inference operator for the Mamdani–Assilian case, and that the defuzzification strategy* $F$ *is the "mean of max" method. Then it is sufficient for the* $(F, \mathcal{D})$*-stability of* $\Theta_{\mathcal{D}}$ *to have satisfied*

$$\mathsf{hgt}\Big(\bigcup_{j=1, j\neq k}^{n} \theta_k(A_j)\Big) < \mathsf{hgt}(\theta_k(A_k)) \tag{67}$$

*for all* $k = 1, \dots, n$.

The proof follows from the corresponding definitions by straightforward routine calculations, and $\mathsf{hgt}$ means the “height” of a fuzzy set, i.e. the supremum of its membership degrees.

Further investigations into this topic are necessary.

## 7 Conclusion

Abstract mathematical tools like mathematical fuzzy logics, like fuzzy set theory and fuzzy relational equations, but also like interpolation strategies, and like operator equations which use the more recent topic of aggregation operators shed interesting light on the formal properties of granular modeling approaches which use the technique of linguistic variables and linguistic rules.

This point of view has been exemplified using considerations upon the fuzzy control related topic of the realizability behavior of systems of linguistic rules, mostly with the background idea to refer to the compositional rule of inference in implementing these rules.

Our reference to the mathematical fuzzy logic $\mathsf{BL}$, with its algebraic semantics determined by the class of all prelinear, divisible, and integral residuated lattices, also opens the way for further generalizations of our considerations toward more general membership degree structures for fuzzy sets: like the type-2 or interval type-2 cases. And the reader should also have in mind that there is a close relationship between rough sets and $L$-fuzzy sets over a suitable 3-element lattice.

Additionally it is easy to recognize that the discussions of the interpolating behavior in Section 5 as well as the operator oriented considerations of Section 6 are essentially independent of the particular choice of the membership degree structure. Hence they too may be generalized to other membership degree setting discussed in this book.

## References

1. Běhounek, L., Bodenhofer, U., Cintula, P.: Relations in fuzzy class theory: initial steps. Fuzzy Sets and Systems 159, 1729–1772 (2008)
2. Calvo, T., Mayor, G., Mesiar, R. (eds.): Aggregation Operators: New Trends and Applications. Physica-Verlag, Heidelberg (2002)
3. Chang, C.C.: Algebraic analysis of many valued logics. Transactions American Mathematical Society 88, 476–490 (1958)
4. Cignoli, R., Esteva, F., Godo, L., Torrens, A.: Basic fuzzy logic is the logic of continuous t-norms and their residua. Soft Computing 4, 106–112 (2000)
5. Cignoli, R.L.O., D’Ottaviano, I.M.L., Mundici, D.: Algebraic foundations of many-valued reasoning. Trends in Logic–Studia Logica Library, vol. 7. Kluwer Academic Publishers, Dordrecht (2000)
6. Nola, A.D., Sessa, S., Pedrycz, W., Sanchez, E.: Fuzzy relation equations and their applications to knowledge engineering. Theory and Decision Library. Series D: System Theory, Knowledge Engineering and Problem Solving, vol. 3. Kluwer Academic Publishers Group, Dordrecht (1989)

7. Dubois, D., Prade, H.: On the use of aggregation operations in information fusion processes. Fuzzy Sets and Systems 142, 143–161 (2004)
8. Dummett, M.: A propositional calculus with denumerable matrix. Journal Symbolic Logic 24, 97–106 (1959)
9. Esteva, F., Godo, L.: Monoidal t-norm based logic: toward a logic for left-continuous t-norms. Fuzzy Sets and Systems 124, 271–288 (2001)
10. Esteva, F., Godo, L., Montagna, F.: Equational characterization of the subvarieties of BL generated by t-norm algebras. Studia Logica 76, 161–200 (2004)
11. Gödel, K.: Zum intuitionistischen Aussagenkalkül. Anzeiger Akad. Wiss. Wien, math.-naturwiss. Kl. 69, 65–66 (1932)
12. Gottwald, S.: Criteria for noninteractivity of fuzzy logic controller rules. In: Straszak, A. (ed.) Large Scale Systems: Theory and Applications 1983 (Warsaw, 1983), pp. 229–233. Pergamon Press, Oxford (1984)
13. Gottwald, S.: Characterizations of the solvability of fuzzy equations. Elektron. Informationsverarbeitung Kybernetik 22, 67–91 (1986)
14. Gottwald, S.: Fuzzy Sets and Fuzzy Logic. Artificial Intelligence. Vieweg, Braunschweig/Wiesbaden, and Tecnea, Toulouse (1993)
15. Gottwald, S.: Generalized solvability behaviour for systems of fuzzy equations. In: Novák, V., Perfilieva, I. (eds.) Discovering the World with Fuzzy Logic, pp. 401–430. Physica-Verlag, Heidelberg (2000)
16. Gottwald, S.: A Treatise on Many-Valued Logics. Studies in Logic and Computation, vol. 9. Research Studies Press, Baldock (2001)
17. Gottwald, S.: Mathematical fuzzy control. A survey of some recent results. Logic Journal of the IGPL 13, 525–541 (2005)
18. Gottwald, S.: On a generalization of fuzzy relation equations. In: Proc. 11th Internat. Conf. IPMU, Paris, vol. 2, pp. 2572–2577, Edition EDK (2006)
19. Gottwald, S.: An abstract approach toward the evaluation of fuzzy rule systems. In: Wang, P.P., Ruan, D., Kerre, E.E. (eds.) Fuzzy Logic. A Spectrum of Theoretical & Practical Issues. Studies in Fuzziness and Soft Computing, vol. 215, pp. 281–297. Springer, Berlin (2007)
20. Gottwald, S.: Calculi of information granules: Fuzzy relational equations. In: Pedrycz, W., Skowron, A., Kreinovich, V. (eds.) Handbook of Granular Computing, ch. 11, pp. 225–248. J. Wiley & Sons, Chichester (2008)
21. Gottwald, S., Novák, V., Perfilieva, I.: Fuzzy control and t-norm-based fuzzy logic. Some recent results. In: Proc. 9th Internat. Conf. IPMU, Annecy, ESIA – Université de Savoie, pp. 1087–1094 (2002)
22. Hájek, P.: Metamathematics of Fuzzy Logic. Trends in Logic, vol. 4. Kluwer Academic Publishers, Dordrecht (1998)
23. Hájek, P., Godo, L., Esteva, F.: A complete many-valued logic with product-conjunction. Archive Math. Logic 35, 191–208 (1996)
24. Haniková, Z.: Standard algebras for fuzzy propositional calculi. Fuzzy Sets and Systems 124, 309–320 (2001)
25. Haniková, Z.: Mathematical and Metamathematical Properties of Fuzzy Logic. PhD thesis, Charles University, Prague, Czech Republic (2003)
26. Hüsam, P.: FATI versus FITA. Ein abstrakter Vergleich gängiger Strategien bei “fuzzy control”-Systemen. Diplomarbeit, Mathematisches Institut, Universität Leipzig (2008)
27. Jenei, S., Montagna, F.: A proof of standard completeness for Esteva and Godo’s logic MTL. Studia Logica 70, 183–192 (2002)

28. Klawonn, F.: Fuzzy points, fuzzy relations and fuzzy functions. In: Discovering the world with fuzzy logic. Stud. Fuzziness Soft Comput, vol. 57, pp. 431–453. Physica, Heidelberg (2000)
29. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms. Kluwer Academic Publishers, Dordrecht (2000)
30. Duncan Luce, R.: A note on Boolean matrix theory. Proc. Amer. Math. Soc. 3, 382–388 (1952)
31. Łukasiewicz, J., Tarski, A.: Untersuchungen über den Aussagenkalkül. Comptes Rendus Séances Soc. Sciences et Lettres Varsovie, cl. III 23, 30–50 (1930)
32. Mamdani, A., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. Internat. J. Man-Machine Studies 7, 1–13 (1975)
33. Novák, V., Perfilieva, I., Gottwald, S.: Fuzzy relation equations via basic predicate fuzzy logic. In: Da Ruan, P., Kerre, E.E. (eds.) Computational Intelligent Systems for Applied Research, pp. 81–88. World Scientific, New Jersey (2002)
34. Perfilieva, I., Tonis, A.: Compatibility of systems of fuzzy relation equations. Internat. J. of General Systems 29, 511–528 (2000)
35. Sanchez, E.: Resolution of composite fuzzy relation equations. Information and Control 30, 38–48 (1976)
36. Zadeh, L.A.: Outline of new approach to the analysis of complex systems and decision processes. IEEE Trans. Systems, Man, and Cybernet. SMC-3, 28–44 (1973)

# Type-2 Fuzzy Logic and the Modelling of Uncertainty in Applications

Robert John and Simon Coupland

De Montfort University, The Gateway, Leicester, LE1 9BH, UK
rij@dmu.ac.uk, simonc@dmu.ac.uk

**Abstract.** Most real world applications contain high levels of uncertainty and imprecision. Sources of the imprecision include sensor noise; variation in actuator performance; linguistic variation between people; temporal modification of expert opinion; and disagreement between experts. Type-2 fuzzy logic is now accepted as a mature technology for coping with this wide variety of sources of uncertainty. This Chapter provides an overview of type-2 fuzzy logic systems providing the reader with an insight into how the various algorithms provide different approaches to modelling uncertainty. We place in context these issues by discussing a number of real world applications that have successfully deployed type-2 fuzzy logic.

## 1 Introduction

Recently there has been significant growth in interest in type-2 fuzzy logic. Type-2 fuzzy logic is an extension of type-1 (regular) fuzzy logic where the membership grade in a fuzzy set is itself measured as a fuzzy number.

Fuzzy sets (Zadeh [57]) have, over the past forty years, laid the basis for a successful method of modelling uncertainty, vagueness and imprecision in a way that no other technique has been able. The use of fuzzy sets in real computer systems is extensive, particularly in consumer products and control applications.

Zadeh [62] presents a powerful argument for the use of fuzzy logic for manipulating perceptions. As has been discussed, his argument is that perceptions (for example, perceptions of size, safety, health and comfort) cannot be modelled by traditional mathematical techniques and that fuzzy logic is more suitable. The discussion about perception modelling is both new and exciting. We argue that type-2 fuzzy sets, since they have non-crisp fuzzy membership functions (that is they are not exact), can model these perceptions more effectively than type-1 fuzzy sets where the membership grades are crisp in nature.

So, we take the position that although fuzzy logic has many successful applications there are a number of problems with the 'traditional' fuzzy logic approach that require a different set of fuzzy tools and techniques for modelling high levels of uncertainty. In particular the argument presented here is that fuzzy logic, as it is commonly used, is essentially **precise** in nature and that for many applications it is unable to model knowledge from an expert adequately. We argued that the modelling of imprecision can be enhanced by the use of type-2 fuzzy sets - providing a higher level of imprecision.

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 185–201.
springerlink.com

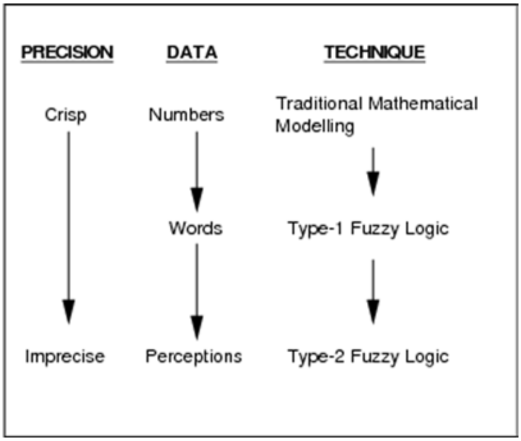


**Fig. 1.** Relationships between imprecision, data and fuzzy technique

Indeed, the tenet of this work is that the success of fuzzy logic can be built on by type-2 fuzzy sets and taken into the next generation of (type-2) fuzzy systems. The use of type-2 fuzzy sets allows for a better representation of uncertainty and imprecision in particular applications and domains. This argument is presented with the use of a mobile robot control application.

The more imprecise or vague the data is, then type-2 fuzzy sets offer a significant improvement on type-1 fuzzy sets. Figure 1 shows the view taken in this work of the relationships between levels of imprecision, data and technique. As the level of imprecision increases then type-2 fuzzy logic provides a powerful paradigm for potentially tackling the problem. Problems that contain crisp, precise data do not, in reality, exist. However some problems can be tackled effectively using mathematical techniques where the assumption is that the data is precise. Other problems (for example, in control) use imprecise terminology that can often be effectively modelled using type-1 fuzzy sets. Perceptions, it is argued here, are at a higher level of imprecision and type-2 fuzzy sets can effectively model this imprecision.

Section 2 provides an overview of type-2 fuzzy sets and type-2 fuzzy logic. The next Section in this chapter, Section 3, presents the history of the field of type-2 fuzzy logic including the recent emergence of generalised type-2 fuzzy system as a viable technology. Section 4 presents the application of type-1, type-2 interval and generalised type-2 fuzzy logic to a mobile robot control application. This example application demonstrates the potential of generalised type-2 fuzzy logic to give an improved performance over type-2 interval fuzzy logic. Section 5 draws conclusions from this work. We also note that some of this material is contained in John and Coupland [15].

## 2 Type-2 Fuzzy Sets and Type-2 Fuzzy Logic

Type-2 fuzzy sets (originally introduced by Zadeh [59]) have membership grades that are fuzzy. That is, instead of being in [0,1] the membership grades are themselves (type-1) fuzzy sets. Karnik and Mendel [25][page 2] provide this definition of a type-2 fuzzy set:

*A type-2 fuzzy set is characterised by a fuzzy membership function, i.e. the membership value (or membership grade) for each element of this set is a fuzzy set in [0,1], unlike a type-1 fuzzy set where the membership grade is a crisp number in [0,1].*

The characterisation in this definition of type-2 fuzzy sets uses the notion that type-1 fuzzy sets can be thought of as a first order approximation to uncertainty and, therefore, type-2 fuzzy sets provide a second order approximation. They play an important role in modelling uncertainties that exist in fuzzy logic systems[21] and are becoming increasingly important in the goal of 'Computing with Words'[61] and the 'Computational Theory of Perceptions'[62].

We now define various terms that relate to type-2 fuzzy sets and state the Representation Theorem (for a detailed discussion and proof of the Representation Theorem the reader is referred to [40]). The first definition we give is a formal definition of a type-2 fuzzy set.

**Definition 1.** *A type-2 fuzzy set, $\tilde{A}$, is characterised by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$*

$$\tilde{A} = \{((x,u), \mu_{\tilde{A}}(x,u)) \mid \forall x \in X, \forall u \in J_x \subseteq [0,1]\} \tag{1}$$

For any given $x$ the $\mu_{\tilde{A}}(x,u)$, $\forall u \in J_x$, is a type-1 membership function as discussed in the introduction. It is this 'extra fuzziness' that is the attraction of type-2 fuzzy sets[21].

It can be seen from this definition that a type-2 membership function is three dimensional. To illustrate this Figure 2 provides an example of type-2 fuzzy set.

We have a three dimensional figure with the axes being $x$, $u$ and $\mu_{\tilde{A}}(x,u)$. The 'spikes' are in [0,1] and represent $\mu_{\tilde{A}}(x,u)$ for a given $(x,u)$. For a given $x$ we have a vertical slice that we call a *secondary membership function*.

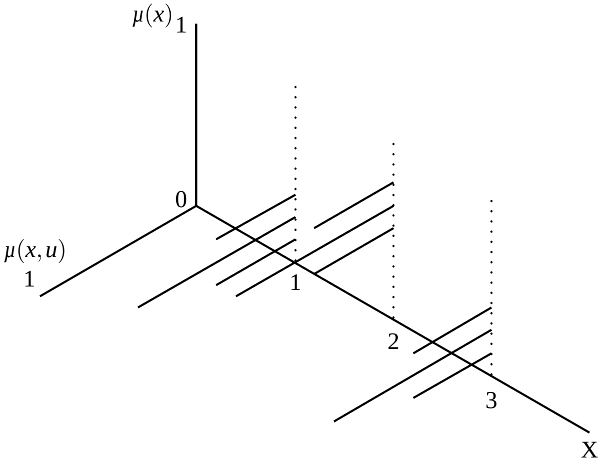


**Fig. 2.** A Type-2 Fuzzy Set

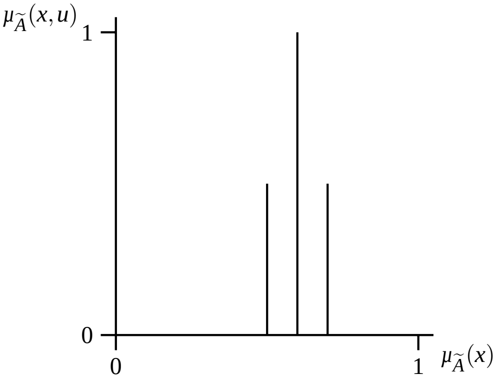


**Fig. 3.** A Secondary Membership Function

**Definition 2.** *At each value x (say $x'$) then $\mu_{\tilde{A}}(x',u)$ is a secondary membership function of $\mu_{\tilde{A}}(x,u)$. We also know this as a vertical slice.*

An example secondary membership function of the type-2 fuzzy set in Figure 2 is given in Figure 3. In this case the secondary membership function is for $x' = 2$ with $\mu_{\tilde{A}}(2) = 0.5/0.5 + 1.0/0.6 + 0.5/0.7$. Note, then, that the type-2 fuzzy set $\tilde{A}$ is the union of all the secondary membership functions. The Representation Theorem lays the basis for the proof of the extended sup-star composition and this, in turn, relies on the notion of an embedded type-2 fuzzy set [40].

**Definition 3.** *For discrete universes of discourse X and U, an embedded type-2 fuzzy set $\tilde{A}_e$ has N elements, where $\tilde{A}_e$ contains exactly one element from $J_{x_1}, J_{x_2}, \ldots, J_{x_N}$, namely $u_1, u_2, \ldots, u_N$, each with its associated secondary grade $f_{x_i}(u_i)$ $(i = 1, \ldots, N)$, i.e.*

$$\tilde{A}_e = \sum_{i=1}^{N} [f_{x_i}(u_i)/u_i]/x_i \qquad u_i \in J_{x_i} \subseteq U = [0,1] \tag{2}$$

Figure 4 gives an example of an embedded type-2 fuzzy set. As can be seen we now have what we might call a ‘wavy slice’ where we have one element (only) from each vertical slice contained in the embedded type-2 fuzzy set.

### 2.1 The Representation Theorem

The definitions so far in this chapter provide enough detail to understand the Representation Theorem[40] needed for this new proof of the extended sup-star composition. We give the Theorem without proof.

**Theorem.** *Let $\tilde{A}_e^j$ denote the jth type-2 embedded fuzzy set for type-2 fuzzy set $\tilde{A}$.*

$$\tilde{A}_e^j \equiv \{(u_i^j, f_{x_i}(u_i^j)), i = 1, \ldots, N\} \tag{3}$$

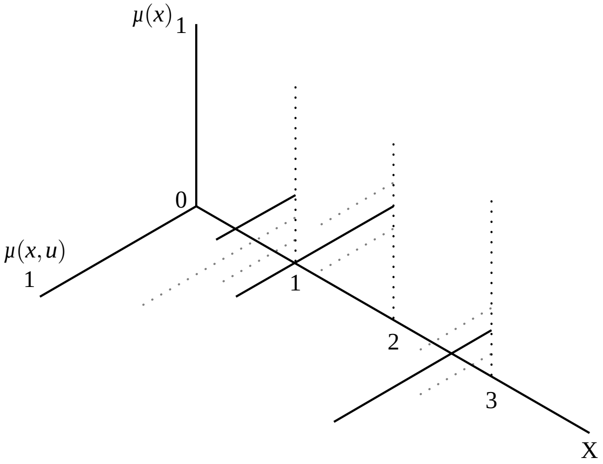


**Fig. 4.** An Embedded Type-2 Fuzzy Set

*where*

$$u_i^j \in \{u_{ik}, k = 1, \ldots, M_i\} \tag{4}$$

*$\tilde{A}$ can be represented as the union of all its type-2 embedded fuzzy sets. That is:*

$$\tilde{A} = \sum_{j=1}^{n} \tilde{A}_e^j \tag{5}$$

*where*

$$n \equiv \prod_{i=1}^{N} M_i \tag{6}$$

We are able to show, then, that a type-2 fuzzy set $\tilde{A}$ is the union of all its type-2 embedded fuzzy sets. This Theorem has allowed for the derivation of union, intersection and complement of type-2 fuzzy sets without use of the extension principle [40]. The union and intersection of embedded type-2 fuzzy sets are as follows. Suppose we have two embedded type-2 fuzzy sets $\tilde{A}_e^j$ and $\tilde{B}_e^i$. The secondary grades at $x_l$ are denoted as $f_{x_l}(u_l^j)$ and $g_{x_l}(w_l^j)$ respectively then

$$\tilde{A}_e^j \cup \tilde{B}_e^i \equiv [F_{x_1}(u_1^j, w_1^i)/u_1^j \vee w_1^i]/x_1 + \ldots + [F_{x_N}(u_N^j, w_N^i)/u_N^j \vee w_N^i]/x_N \tag{7}$$

where, for each $l = 1, \ldots, N$,

$$F_{x_l}(u_l^j, w_l^i) = h[f_{x_l}(u_l^j), g_{x_l}(w_l^i)]$$

and $h$ is a t-tnorm. This also known as the join ($\sqcup$). So that

$$\mu_{\tilde{A}_e^j} \sqcup \mu_{\tilde{B}_e^i} \equiv [F_{x_1}(u_1^j, w_1^i)/u_1^j \vee w_1^i]/x_1 + \ldots + [F_{x_N}(u_N^j, w_N^i)/u_N^j \vee w_N^i]/x_N \tag{8}$$

The intersection is given by

$$\tilde{A}^j_e \cap \tilde{B}^i_e \equiv [F_{x_1}(u^j_1, w^i_1)/u^j_1 \wedge w^i_1]/x_1 + \ldots + [F_{x_N}(u^j_N, w^i_N)/u^j_N \wedge w^i_N]/x_N \tag{9}$$

This also known as the meet ($\sqcap$). So that

$$\mu_{\tilde{A}^j_e} \sqcap \mu_{\tilde{B}^i_e} \equiv [F_{x_1}(u^j_1, w^i_1)/u^j_1 \wedge w^i_1]/x_1 + \ldots + [F_{x_N}(u^j_N, w^i_N)/u^j_N \wedge w^i_N]/x_N \tag{10}$$

# 3 The Historical Development of Type-2 Fuzzy Logic

Type-2 fuzzy logic is a growing research topic. In this section we discuss the main themes in type-2 fuzzy logic and highlight some applications.

## 3.1 Type-2 Fuzzy Sets Appear

Type-2 fuzzy sets were first defined and discussed in a trilogy of papers by Zadeh [1975*a*, 1975*b*, 1975*c*]. These papers concentrated on the notion of a fuzzy set where the memberships grades of a fuzzy set are measured with linguistic terms such as *low*, *medium* and *high*. Logical connectives for such sets were also given, although the terms join and meet were not used. Zadeh only explored the use of the minimum and maximum operators t-norm and t-conorm when investigating the logical operations. Mizumoto and Tanaka [1976,1981] and Dubois and Prade [1980] both studied the logical connectives of what became known as secondary membership functions. Mizumoto and Tanaka were the first to use the terms join and meet for these logical connectives. Both Dubois and Prade and Mizumoto and Tanaka studied the join and meet under a variety of t-norm and t-conorm operators.

## 3.2 Type-2 Interval Fuzzy Sets Are Promoted

Turksen [1993,1993*a*,1995], Schwartz [48] and Klir and Folger [29] promoted the use of type-2 fuzzy sets, at that time called interval valued or IV fuzzy sets. Schwartz believes that type-2 interval fuzzy sets should be employed when the linguistic uncertainty of a term cannot be sufficiently modelled by the type-1 methods. Klir and Folger advocate the use of IV fuzzy sets when the membership functions of type-1 fuzzy sets could not be agreed upon. These arguments were explored in greater detail by Mendel [36]. Turksen put forward a collection of logical connectives for type-2 interval fuzzy sets noting that the expressive power of type-2 fuzzy reasoning lies in the ability to retain the uncertainty throughout the inferencing process.

## 3.3 Type-reduction Is Defined

Karnik and Mendel [1998,1998*b*,2001] defined type-reduction, the technique used for defuzzifing type-2 fuzzy sets, by applying the extension principle to a variety of type-1 defuzzifiers. The notion of an output processing stage of a type-2 fuzzy system was developed in these papers.

### 3.4 Type-2 Fuzzy Logic Systems Are Fully Defined

Karnik and Mendel [25, 28] gave a complete description of the fuzzy inferencing process. This allowed work on the application of type-2 fuzzy logic to proceed. Around this time John [1998*a*,1998*b*,1999,1999*a*] published a series of review papers on type-2 fuzzy systems. Early applications of the technology also began to appear (see for example John [19, 23] and Karnik and Mendel [27]).

### 3.5 The First Textbook on the Subject of Type-2 Fuzzy Logic Appears

Following the consolidation of the definitions and existing literature by John and Karnik and Mendel, the field was opened up to a wider potential audience with the publication of the first type-2 textbook. Uncertain Rule-Based Fuzzy Logic System: Introduction and New Directions was written by Mendel [2001*a*] and published in 2001. This textbook references a great deal of the work on type-2 fuzzy logic that had been published to date, bringing together many of Mendel's earlier publications.

### 3.6 The Representation Theorem Is Defined

Mendel and John [40] gave the representation theorem of type-2 fuzzy sets. By representing a type-2 fuzzy set as a collection of simpler type-2 embedded sets it is possible to define operations of type-2 fuzzy sets without the use of the extension principle. The motivation behind this work was that by eliminating the need to learn about the extension principle, the field would be more accessible to type-1 fuzzy practitioners. However, the representation theorem has its own learning curve, and is not significantly simpler to understand than the extension principle. One of the outcomes of the representation theorem has been the definition of arithmetic operators for type-2 fuzzy numbers by Coupland and John [1].

### 3.7 Issues of Computational Complexity Begin to Be Explored

The complexity of join and meet operations and type-reduction of a type-2 fuzzy set limit the applicability of type-2 methods. Although type-2 interval sets are simpler, type-reduction is still a problem, due to inherent complexity and redundancies. The iterative method (Karnik and Mendel [28]) and the Wu-Mendel [55, 56] approximation were developed to make the type-reduction of type-2 interval fuzzy sets more efficient. This has led to the majority of the publications in the field of type-2 only discussing type-2 interval methods. Indeed, many authors refer to type-2 interval fuzzy set as type-2 fuzzy sets and add the qualifying term 'generalised' when discussing actual type-2 fuzzy sets. The computational problems of join and meet were effectively resolved by Karnik and Mendel [28]. This work is also discussed by the author, along with some aspects of the geometric approach in Coupland *et al.* [2, 3]. Greenfield *et al.* [12] give an efficient method for approximating the type-reduced set of a type-2 fuzzy set using a stochastic approach.

### 3.8 Computing with Words Appears

Zadeh [61, 62] made the claim that fuzzy logic, approximately at least, equates to computing with words (CWW). In CWW numbers are replaced with words not only when reasoning, but also when solving calculations. Zadeh‘s examples use fuzzy granules to model words. A fuzzy granule is actually the Footprint Of Uncertainty of a type-2 interval fuzzy set. Both Mendel [37, 39] and Turksen [52] point out that CWW requires type-2 fuzzy sets, both opting to use the simpler type-2 interval representations. Mendel [36] re-emphasised this point by demonstrating that human models of words as obtained through a survey require at least interval representations.

### 3.9 Control Applications

With the iterative method and the Wu-Mendel approximation allowing fast execution of type-2 fuzzy systems, control applications began to emerge. Melin and Castillo [34, 35] used type-2 interval systems in the context of plant control. Hagras [13] demonstrated that a type-2 interval fuzzy logic controller could outperform a type-1 fuzzy controller under large uncertainties. Wu and Tan [54] applied type-2 interval systems to the control of a complex multi-variable liquid level process. Figueroa *et al.* [10] used a type-2 interval control for non-autonomous robots in the context of a robot football game. The authors’ have performed a comprehensive study of both general and type-2 interval fuzzy controllers for an autonomous mobile robot. Some aspects of these studies are presented in Section 4 of this work and in Coupland [3]. Doctor *et al.* [8] used a type-2 interval system to model and adapt to the behaviour of people in an intelligent dormitory room. Additional work on type-2 fuzzy logic hardware has also contributed great to the field. Lynch *et al.* [31, 32] have implemented an industrial type-2 interval control system for large marine diesel engines which has very good performance, both in control response and cycle times. Melgarejo *et al.* [33] have also developed a limited hardware implementation of a type-2 interval controller which has a lesser focus on industrial application. Coupland *et al.* [6] have implemented generalised type-2 fuzzy logic, complete with Integrated Development Environment with a focus on dissemination of the technology.

### 3.10 Medical Applications

Medical applications are one of the few areas where a generalised type-2 fuzzy logic has been used in preference to type-2 interval fuzzy logic. This is largely because such systems do not require fast execution times but do contain large uncertainties. John *et al.* [18, 23] used a type-2 fuzzy system for the pre-processing of tibia radiographic images. Garibaldi *et al.* [11, 46] have done extensive work on assessing the health of a new born baby using knowledge of acid-base balance in the blood from the umbilical cord. Innocent and John [14] proposed the use of fuzzy cognitive maps to aid the differential diagnosis of confusable diseases and suggest that type-2 cognitive maps may yield improved results. Di Lascio *et al.* [7] also used type-2 fuzzy sets to model differential diagnosis of diseases, modelling the compatibility of the symptom to a disease as a linguistic term. John *et al.* [2001,2001*a*] used type-2 fuzzy sets to model the perception of clinical opinions of nursing staff as linguistic terms.

### 3.11 Signal Processing

Signal processing, like control, has to date only used type-2 interval methods. Liang and Mendel [30] implemented a fuzzy adaptive filter for the equalization of non-linear time-varying channels. Mitchell [42] defined a similarity measure for use with type-2 fuzzy sets which was used in a radiographic image classifier. Karnik and Mendel [27] used a type-2 interval system to predict the next value in a chaotic time series. Musikasuwan *et al.* [45] investigated how the learning capabilities of type-1 and type-2 interval systems differ according to the number of learning parameters used. Both systems were designed to to predict a Mackey-Glass time series.

### 3.12 Generalised Type-2 Fuzzy Logic Emerges as a Viable Technology

Two very recent major advances in generalised type-2 fuzzy logic have had a significant impact on the usability of generalised type-2 fuzzy systems. Coupland‘s geometric model [4, 5] of type-2 fuzzy sets and systems have eliminated the historical problem of the computing the centroid of a type-2 fuzzy set. The work presented in the following Section of this chapter is only possible because of the reduction in computation provided by the geometric model. The simultaneous definition of alpha-planes (or z-slices) of type-2 fuzzy sets by Mendel and Liu [41], and Wagner and Hagras [53] give an approximation of the geometric method which is highly efficient and highly parallel. Definitions and investigations of these techniques are currently being undertaken by a number of researchers in the field.

### 3.13 Summary

This Section has given the major developments that have taken place in the field of type-2 fuzzy logic and places them in a historical context. Type-2 literature has become predominately concerned with type-2 interval methods. The likely reason for this is the elimination of the computational problems for type-2 interval methods. The authors' view is that generalised type-2 fuzzy logic has a great deal to offer as will be demonstrated in the following section.

## 4 Type-2 Fuzzy Logic Controllers

This section presents a comparison of three fuzzy logic controller which are given the task of navigating around a curved obstacle. Each of the three controllers is based on a different fuzzy technology:

- Controller 1 uses type-1 fuzzy logic;
- Controller 2 uses type-2 interval fuzzy logic, and
- Controller 3 uses hybrid type-2 fuzzy logic.

The type-1 controller was designed first and provides a basis for controllers 2 and 3. The hybrid type-2 controller makes use of geometric fuzzy logic in order to achieve the execution speeds requires by the robot control system.

### 4.1 Task Selection

There are currently no reported systems (except Coupland *et al.* [2006] which reports some aspects of this experiment) where generalised type-2 fuzzy logic has been applied to a control application. This Section describes the first such application which has been made possible with the introduction of geometric type-2 fuzzy logic. As discussed in earlier, type-2 fuzzy logic systems should be able to cope with the uncertainties inherent in control applications. To best evaluate geometric type-2 fuzzy logic in a control application a difficult mobile robot navigation problem was designed. Type-2 interval fuzzy logic has already been applied to such an application by Hagras [13]. The Hagras study demonstrated improved performance in navigation tasks when using type-2 interval rules rather than type-1 under environmental uncertainties. One of the limitations of the Hagras study was that the robot only performed eight runs and therefore, it is difficult to state the significance, if any, of this performance improvement. However, the Hagras study demonstrated that mobile robot navigation is a useful application area for exploring the potential of type-2 fuzzy logic in control applications.

The task of mobile robot navigation represents a significant challenge for a type-2 FLC. The control system has to operate in real time on limited hardware resources. The environment which the robot has to operate in is challenging. The sensors on the robot are operating in the real world and are prone to noise and error. For example the accuracy of a sonar sensor is likely to be reduced the further away an object is. Background noise in the environment may also effect the sonar reading. The level of traction between the wheels and the floor depends on the type of flooring, type pressures and the speed at which the wheels are moving. The task to be completed by the FLC presented in this chapter is to navigate a mobile robot around the curved edge of a wall like obstacle maintaining a distance of 0.5 metres between the centre of the robot and the obstacle at all times. A diagram of the robot, the obstacle and the ideal path that the robot should follow around the obstacle is given in Figure 5. The initial position of the robot puts the obstacle at a right angle to the left wheel of the robot. The initial distance between the obstacle and the centre of the robot is set at 0.5 metres. The robot is facing the correct direction to begin navigation of the obstacle. This start position places the robot just below the start point of the ideal path that should be taken by the robot around the obstacle. Once the centre of the robot crosses the dotted start line

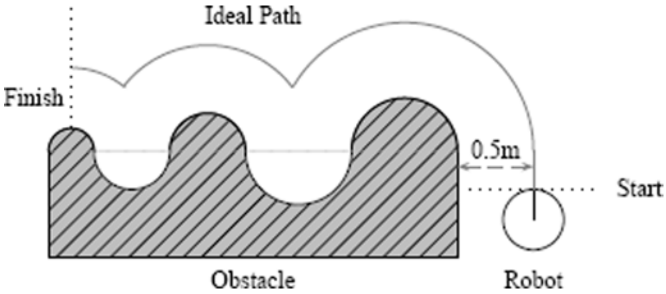


**Fig. 5.** Mobile Robot and Obstacle

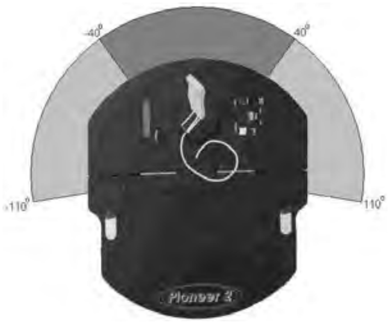


**Fig. 6.** The Pioneer 2 Mobile Robot

tracking begins. Once the centre of the robot crosses the dotted finish line tracking stops. All runs started from the same initial position. The task of the FLC is essentially to minimise the deviation from the ideal path between the start and finish lines.

The controllers were deployed on the commercially available pioneer 2 robot (depicted in Fig 6) built by ActivMedia. The robot has an on board personal computer which the software based FLCs were implemented on. This PC links to a microcontroller which is directly connected to the sensors and actuators. An array of eight sonar sensors each with a range of 3 metres provides sensory capability. Two independently driven wheels give mobility to the robot. The FLCs had four inputs, $d_1$, $\theta_1$, $d_2$ and $\theta_2$:

- The angle $\theta_1$ is the angle to the closest object detected by all eight sensors. $\theta_1$ is given as a value between $-110^o$ and $110^o$;
- the angle $\theta_2$ is the angle to the closest object detected by the middle four sensors. $\theta_2$ takes a value between $-40^o$ and $40^o$;
- the distance $d_1$ is the distance to the nearest object detected by all eight sensors, and
- the distance $d_2$ is the distance to the nearest object detected by the middle sensors.

The only output from the system is the change in direction ($\delta h$) of the heading of robot. Since only the direction of the robot is being altered the speed of the robot is kept constant at $0.1\text{ms}^{-1}$. The robot travels at this speed when moving in a straight line. However, when turning a component of this speed is taken up as rotational velocity. The robot is always moving forwards and can never go backwards.

The Aria software library provided with the robot requires that control commands are executed within a tenth to a quarter of a second window. This is the definition of real time for this robot, command execution within a quarter of a second or the robot operations will shutdown. This is quite a low requirement by control standards. It is however a significant challenge to perform type-2 fuzzy inferencing on limited hardware within a quarter of a second.

## 4.2 Controller Design

At the heart of FLS is the rule base. We started with a type-1 fuzzy rule base and blurred the membership functions. In this work the control rules were based on the experience of a robot operator. Experience of how to drive the robot around the obstacle was gained by the author. A joystick was connected to the robot over a wireless network. This joystick was then used to manoeuvre the robot around the obstacle. This process was repeated until the author was competent at driving the robot around the obstacle. The rules were based on this experience of driving the robot around the obstacle manually with a joystick.

The FLC also used an idea from control theory, the change in error over time, the derivative $\delta e$. This added an element of proportionality to the controller (Reznik [47]). To obtain the value of $\delta e$ the gradient of a best fit line placed through the last four error measurements $e$, where $e = 500 - d_1$ was taken. Taking $\delta e$ is useful as it gives a measure of whether the robot is moving toward the ideal path or away from it. This is particularly useful with this configuration of pioneer robots as they do not have any sonar sensors at the rear to detect whether the robot is moving toward or away from an object.

## 4.3 Results

The path each robot FLC took around the obstacle was tracked fifty times. These tracked paths are depicted in Figures 7, 8 to 9. The error for each point in this tracked path relative to an ideal path was calculated. The RMSE for each tracked run around the obstacle was then calculated. The mean, median, standard deviation and coefficient of variance over the fifty runs was then calculated for each robot FLC. These results are given in table 1.

**Table 1.** The Mean, Median, Standard Deviation and Coefficient of Variance of Error for the Six Robot FLC Over Fifty Runs. All numbers quoted to 4 d.p.

| Controller | Mean Error | Median Error | St Dev of Error | Co Var of Error |
|---|---|---|---|---|
| 1 | 13.5852 | 13.4185 | 1.0995 | 0.0809 |
| 2 | 12.5394 | 11.9779 | 2.0543 | 0.1638 |
| 3 | 9.8171 | 9.7783 | 1.0185 | 0.1038 |

An initial visual comparison would suggest that the controller 3 performed most consistently. Controller 2 had a wide but consistent spread. Controller 1 had spread of paths somewhere between the two with a few paths quite far outside the main spread. It is difficult judge the error of the controllers visually, although the Controller 3 path appear more tightly packed than the other two.

The results from the experiment did not display either normality or equality of variance. Therefore the non-parametric Kruskal-Wallis test was used to assess whether or not there are any differences between the controllers' performance. The test gave a $H$ statistic value of 97.01 and a $p$ value of $< 0.0005$, suggesting strong evidence of differences between the controllers. The Kruskal-Wallis test works by ranking the data by median value. Table 2 gives the median values and the average ranking the three controllers. The median positions and mean rankings do point to the type-2 controller having the best performance, followed by the interval type-2 controller and then the

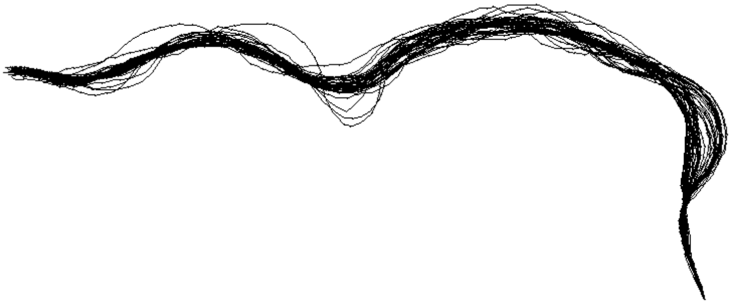


**Fig. 7.** Paths Taken By Controller 1

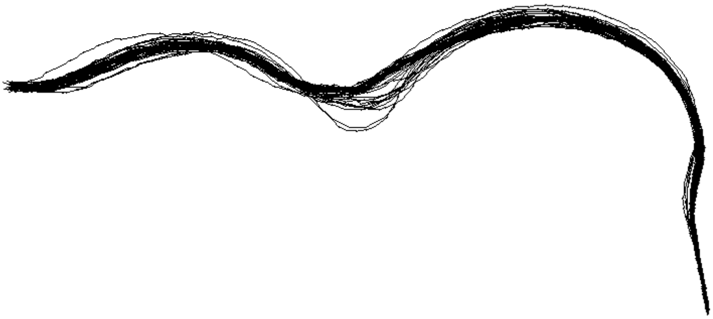


**Fig. 8.** Paths Taken By Controller 2

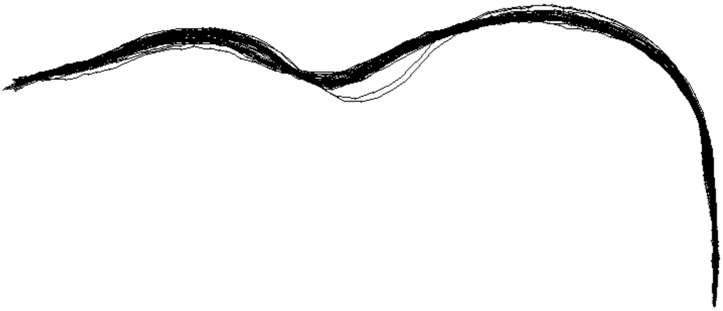


**Fig. 9.** Paths Taken By Controller 3

**Table 2.** The Median and Average Rank of the Three Controllers from the Kruskal-Wallis Test Procedure

| Controller | 1 | 2 | 3 |
|---|---|---|---|
| Median | 13.392 | 11.961 | 9.802 |
| Average Rank | 113.3 | 84.2 | 29.0 |

type-1 controller. This performance ranking is identical to the ordering of the $\overline{RMSE}$ of the FLC. Looking at consistency of performance both the test for equal variances and the values of $\sigma RMSE$ suggest that the type-1 and type-2 FLC were equally consistent. The interval type-2 FLC had a less consistent performance.

It is important to compare the outcomes that are suggested by the statistical comparison with those give by a visual comparison of the paths. The statistics suggest that FLC performance is ranked type-2, then interval type-2 and then type-1. The path depictions support this conclusion. The statistics suggest that the type-1 and type-2 FLC were equal in the consistency of performance. This is not immediately clear from the visual comparison. Take into account that the type-1 FLC gave the worst performance. A view can be taken that the type-1 FLC made more errors, however these errors were made consistently. The type-2 interval FLC gave a middling performance, but on occasionally made significant errors. This relates well to the visual paths. To summarise these points:

- The type-2 FLC performed consistently well.
- The interval type-2 FLC performed quite well, but was a little inconsistent.
- The type-1 FLC performed relatively badly, but was consistent in this level of error.

These findings are supported by a visual inspection of taken and by a statistical analysis of those paths.

## 5 Conclusion

This Chapter has presented an introduction to type-2 fuzzy sets, an overview of the history of type-2 fuzzy logic with emphasis on applications and a detailed description of a real application in control. There is much work still to be done on type-2 fuzzy logic and we believe the applications and theoretical results will continue to grow.

## References

1. Coupland, S., John, R.: An Approach to Type-2 Fuzzy Arithmetic. In: Proc. UK Workshop on Computational Intelligence, pp. 107–114 (2004)
2. Coupland, S., John, R.: Fuzzy Logic and Computational Geometry. In: Proc. RASC 2004, Nottingham, England, December 2004, pp. 3–8 (2004)
3. Coupland, S., John, R.: Towards More Efficient Type-2 Fuzzy Logic Systems. In: Proc. FUZZ-IEEE 2005, Reno, NV, USA, May 2005, pp. 236–241 (2005)
4. Coupland, S., John, R.: New Geometric Inference Techniques for Type-2 Fuzzy Sets. International Journal of Approximate Reasoning 49(1), 198–211 (2008)
5. Coupland, S., Wheeler, J., Gongora, M.: A generalised type-2 fuzzy logic system embedded board and integrated development environment. In: Proc. FUZZ-IEEE 2008 (in WCCI 2008), Hong Kong (accepted for publication, 2008)
6. Coupland, S., John, R.: Geometric logical operations for type-2 fuzzy sets. In: Proc. IPMU 2008, Malaga (submitted, December 2007)
7. Di Lascio, L., Gisolfi, A., Nappi, A.: Medical differential diagnosis through Type-2 Fuzzy Sets. In: Proc. FUZZ-IEEE 2005, Reno, NV, USA, May 2005, pp. 371–376 (2005)

8. Doctor, F., Hagras, H., Callaghan, V.: A Type-2 Fuzzy Embedded Agent For Ubiquitous Computing Environments. In: Proc. FUZZ-IEEE 2004, Budapest, Hungary, July 2004, pp. 1105–1110 (2004)
9. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
10. Figueroa, J., Posada, J., Soriano, J., Melgarejo, M., Rojas, S.: A Type-2 Fuzzy Controller for Tracking Mobile Objects in the Context of Robotic Soccer Games. In: Proc. FUZZ-IEEE 2005, Reno, AZ, USA, May 2005, pp. 359–364 (2005)
11. Garibaldi, J.M., Westgate, J.A., Ifeachor, E.C., Greene, K.R.: The Development and Implementation of an Expert System for the Analysis of Umbilical Cord Blood. Artificial Intelligence in Medicine 10(2), 129–144 (1997)
12. Greenfield, S., John, R., Coupland, S.: A Novel Sampling Method for Type-2 Defuzzification. In: Proc. UKCI 2005, pp. 120–127 (2005)
13. Hagras, H.: A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots. IEEE Transactions on Fuzzy Systems 12, 524–539 (2004)
14. Innocent, P., John, R.I.: Computer Aided Fuzzy Medical Diagnosis. Information Sciences 162, 81–104 (2004)
15. John, R., Coupland, S.: Type-2 Fuzzy Logic: A Historical View. IEEE Computational Intelligence Magazine 2(1), 57–62 (2007)
16. John, R., Lake, S.: Modelling nursing perceptions using type-2 fuzzy sets. In: EUROFUSE 2001 Workshop on Preference Modelling and Applications, pp. 241–246 (2001)
17. John, R.I.: Type-2 Fuzzy Sets. Expert Update, 2(2) (1999) ISSN 1465-4091
18. John, R.I., Innocent, P.R., Barnes, M.R.: Neuro-fuzzy clustering of radiographic tibia image data using type-2 fuzzy sets. Information Sciences 125, 203–220 (2000)
19. John, R.I.: Type-2 inferencing and community transport scheduling. In: Proc. Fourth European Congress on Intelligent Techniques and Soft Computing, EUFIT 1996, Aachen, Germany, September 1996, p. 1369 (1996)
20. John, R.I.: Type–2 Fuzzy Sets for Knowledge Representation and Inferencing. In: Proc. 7th Intl. Conf. on Fuzzy Systems FUZZ-IEEE 1998, pp. 1003–1008 (1998)
21. John, R.I.: Type 2 Fuzzy Sets: An Appraisal of Theory and Applications. International Journal of Uncertainty, Fuzziness and Knowledge Based Systems 6(6), 563–576 (1998)
22. John, R.I.: Fuzzy sets of type-2. Journal of Advanced Computational Intelligence 3(6), 499–508 (1999)
23. John, R.I., Innocent, P.R., Barnes, M.R.: Type–2 Fuzzy Sets and Neuro-Fuzzy Clustering of Radiographic Tibia Images. In: Proc. FUZZ-IEEE 1998, pp. 1373–1376 (1998)
24. John, R.I., Lake, S.: Type-2 fuzzy sets for modelling nursing intuition. In: Proc. Joint 9th IFSA World Congress and 20th NAFIPS International Conference, July 2001, pp. 1920–1925 (2001)
25. Karnik, N.N., Mendel, J.M.: Introduction to Type-2 Fuzzy Logic Systems. In: Proc. IEEE World Congress on Computational Intelligence, Anchorage, Alaska, USA, pp. 915–920 (1998)
26. Karnik, N.N., Mendel, J.M.: Type-2 Fuzzy Logic Systems: Type-Reduction. In: Proc. IEEE Systems, Man and Cybernetics, pp. 2046–2051 (1998)
27. Karnik, N.N., Mendel, J.M.: Application of Type-2 Fuzzy Logic System to Forecasting of Time-Series. Information Sciences 120, 89–111 (1999)
28. Karnik, N.N., Mendel, J.M.: Centroid of a type-2 fuzzy Set. Information Sciences 132, 195–220 (2001)
29. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, Englewood Cliffs (1988)
30. Liang, Q., Mendel, J.M.: Equalization of Nonlinear Time-Varying Channels Using Type-2 Fuzzy Adaptive Filters. IEEE Transactions on Fuzzy Systems 8, 551–563 (2000)

31. Lynch, C., Hagras, H., Callaghan, V.: Embedded Type-2 FLC for Real-Time Speed Control of Marine and Traction Diesel Engines. In: Proc. FUZZ-IEEE 2005, Reno, AZ, USA, May 2005, pp. 347–352 (2005)
32. Lynch, C., Hagras, H., Callaghan, V.: Parallel type-2 fuzzy logic co-processors for engine management. In: Proc. FUZZ-IEEE 2007, London, pp. 907–912 (2007)
33. Melgarejo, M., Pena-Reyes, C.: Hardware Architecture and FPGA Implementation of a Type-2 Fuzzy System. In: Proc. GLSVSLI 2004, Boston, Massachusetts, USA, April 2004, pp. 458–461 (2004)
34. Melin, P., Castillo, O.: Fuzzy Logic for Plant Monitoring and Diagnostics. In: Proc. NAFIPS 2003, July 2003, pp. 20–25 (2003)
35. Melin, P., Castillo, O.: Intelligent Control of Non-Linear Dynamic Plants Using Type-2 Fuzzy Logic and Neural Networks. In: Proc. FUZZ-IEEE 2004, Budapest, Hungary (July 2004)
36. Mendel, J.M.: Computing With Words, When Words Mean Different Things to Different People. In: Proc. of Third International ICSC Symposium on Fuzzy Logic and Applications, Rochester Univ., Rochester (1999)
37. Mendel, J.M.: The Perceptual Computer: an Architecture for Computing With Words. In: Proc. FUZZ-IEEE 2001, Melbourne, Australia (2001)
38. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
39. Mendel, J.M.: Fuzzy sets for words: a new beginning. In: Proc. FUZZ-IEEE 2003, St. Louis, MO, USA, pp. 37–42 (2003)
40. Mendel, J.M., John, R.I.: Type-2 Fuzzy Sets Made Simple. IEEE Transaction on Fuzzy Systems 10(2), 117–127 (2002)
41. Mendel, J.M., Liu, F.: On new quasi-type-2 fuzzy logic systems. In: FUZZ-IEEE 2008, Hong Kong (June 2008)
42. Mitchell, H.B.: Pattern Recognition Using Type-II Fuzzy Sets. Information Sciences 170, 409–418 (2005)
43. Mizumoto, M., Tanaka, K.: Some properties of fuzzy set of type-2. Information and control 31, 312–340 (1976)
44. Mizumoto, M., Tanaka, K.: Fuzzy Sets of Type 2 Under Algebraic Product and Algebraic Sum. Fuzzy Sets and Systems 5, 277–290 (1981)
45. Musikasuwan, S., Ozen, T., Garibaldi, J.M.: An investigation into the effect of number of model parameters on performance in type-1 and type-2 fuzzy logic systems. In: Proc. 10th Information Processing and Management of Uncertainty in Knowledge Based Systems (IPMU 2004), Perugia, Italy, pp. 1593–1600 (2004)
46. Ozen, T., Garibaldi, J.M.: Investigating Adaptation in Type-2 Fuzzy Logic Systems Applied to Umbilical Acid-Base Assessment. In: Proc. of the 2003 European Symposium on Intelligent Technologies, Oulu, Finland, July 2003, pp. 289–294 (2003)
47. Reznik, L.: Fuzzy Controllers. Reed Elsevier (1997)
48. Daniel, G.: Schwartz. The case for an interval-based representation of linguistic truth. Fuzzy Sets and Systems 17, 153–165 (1985)
49. Türkşen, I.B.: Interval-valued fuzzy sets and fuzzy connectives. Interval Computations 4, 35–38 (1993)
50. Türkşen, I.B.: Interval-valued fuzzy uncertainty. In: Proc. Fifth IFSA World Congress, Seoul, Korea, July 1993, pp. 35–38 (1993)
51. Türkşen, I.B.: Knowledge representation and approximate reasoning with type ii fuzzy sets. In: Proc. FUZZ-IEEE 1995, Yokohama, Japan, March 1995, vol. 2, pp. 1911–1917 (1995)
52. Türkşen, I.B.: Type 2 Representation and Reasoning for CWW. Fuzzy Sets and Systems 127, 17–36 (2002)

53. Wagner, C., Hagras, H.: zslices - towards bridging the gap between interval and general type-2 fuzzy logic. In: FUZZ-IEEE 2008, Hong Kong (June 2008)
54. Wu, D., Tan, W.W.: A Type-2 Fuzzy Logic Controller for the Liquid-level Process. In: Proc. FUZZ-IEEE 2004, Budapest, Hungary, July 2004, pp. 953–958 (2004)
55. Wu, H., Mendel, J.M.: Introduction to Uncertainty Bounds and Their Use in the Design of Interval Type-2 Fuzzy Logic Systems. In: Proc. FUZZ-IEEE 2001, Melbourne, Australia (2001)
56. Wu, H., Mendel, J.M.: Uncertainty bounds and their use in the design of interval type-2 fuzzy logic systems. IEEE Transactions on Fuzzy Systems 10, 622–639 (2002)
57. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
58. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning – I. Information Sciences 8, 199–249 (1975)
59. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning – II. Information Sciences 8, 301–357 (1975)
60. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning – III. Information Sciences 9, 43–80 (1975)
61. Zadeh, L.A.: Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems 4, 103–111 (1996)
62. Zadeh, L.A.: From Computing with Numbers to Computing with Words – From Manipulation of Measurements to Manipulation of Perceptions. IEEE Transactions on Circuits and Systems – I:Fundamental Theory and Applications 45, 105–119 (1999)

# Interval Type-2 Fuzzy Logic Applications

Oscar Castillo and Patricia Melin

Tijuana Institute of Technology,
Division of Graduate Studies,
Tijuana, Mexico
{ocastillo,epmelin}@hafsamx.org

**Abstract.** In this chapter three applications of interval type-2 fuzzy logic are considered. First, we consider the use of interval type-2 fuzzy systems in conjunction with modular neural networks for image recognition. A type-2 fuzzy system is used for feature extraction in the training data, and another type-2 fuzzy system is used to find the optimal parameters for the integration method of the modular neural network. Type-2 Fuzzy Logic is shown to be a tool to help improve the results of a neural system by facilitating the representation of the human perception. The second application involves edge detection in digital images, which is a problem that has been solved by means of the application of different techniques from digital signal processing, and also the combination of some of these techniques with type-1 fuzzy systems have been proposed. In this chapter a new interval type-2 fuzzy method is implemented for the detection of edges and the results of three different techniques for the same goal are compared. The third application, concerns the problem of stability, which is one of the more important aspects in the traditional knowledge of Automatic Control. Interval type-2 fuzzy logic is an emerging and promising area for achieving intelligent control (in this case, Fuzzy Control). In this chapter we use the Fuzzy Lyapunov Synthesis, as proposed by Margaliot, to build a Lyapunov stable type-1 fuzzy logic control system, and then we make an extension from a type-1 to a type-2 fuzzy controller, ensuring the stability on the control system and proving the robustness of the corresponding fuzzy controller.

## 1 Interval Type-2 Fuzzy Logic for Image Recognition

At the moment, many methods for image recognition are available. But most of them include a phase of feature extraction or another type of preprocessing closely related to the type of image to recognize (Melin and Castillo, 2005) (Chuang et al., 2000). The method proposed in this paper can be applied to any type of images, because the preprocessing phase does not need specific data about the type of image (Melin et al., 2007) (Mendoza and Melin, 2007).

Even if the method was not designed only for face recognition, we have made the tests with the ORL face database (AT&T Laboratories Cambridge) composed of 400 images of size 112x92. There are 40 persons, with 10 images of each person. The images are taken at different times, lighting and facial expressions. The faces are in upright position of frontal view, with slight left-right rotation. Figure 1 shows the 10 samples of one person in ORL database. To explain the proposed steps of the method, we need to separate it them in two phases: the training phase in figure 3 and the recognition phase in figure 4.

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 203 – 231.
springerlink.com

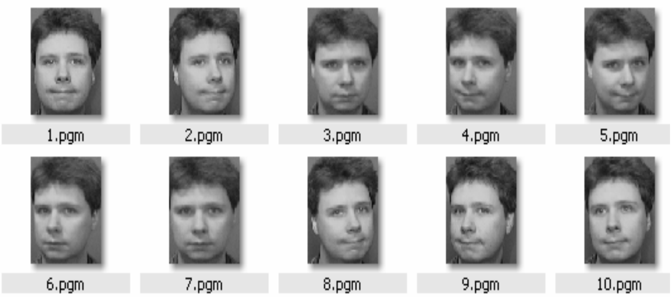


**Fig. 1.** Set of 10 samples of a person in ORL

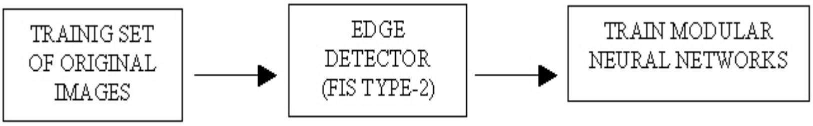

TRAINIG SET OF ORIGINAL IMAGES
EDGE DETECTOR (FIS TYPE-2)
TRAIN MODULAR NEURAL NETWORKS

**Fig. 2.** Steps in Training Phase

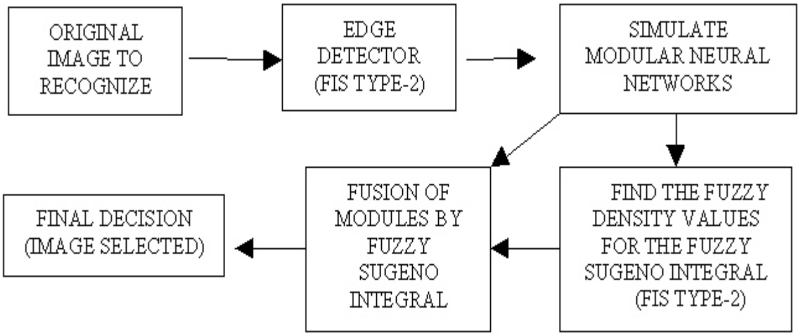

ORIGINAL IMAGE TO RECOGNIZE
EDGE DETECTOR (FIS TYPE-2)
SIMULATE MODULAR NEURAL NETWORKS
FIND THE FUZZY DENSITY VALUES FOR THE FUZZY SUGENO INTEGRAL (FIS TYPE-2)
FUSION OF MODULES BY FUZZY SUGENO INTEGRAL
FINAL DECISION (IMAGE SELECTED)

**Fig. 3.** Steps in Recognition Phase

## 2 Type-2 Fuzzy Inference System as an Edge Detector

In previous work we presented an efficient Fuzzy Inference System for edges detection, in order to use the output image like input data for modular neural networks (Mendoza and Melin, 2006). In the proposed technique, it is necessary to apply Sobel operators to the original images, then use a Fuzzy Inference System Type-2 to generate the vector of edges that would serve like input data in a neural network. Type-2 Fuzzy Logic enables us to handle uncertainties in decision making and recognition in a more convenient way and for this reason was proposed (Castillo et al., 2007).

For the Type-2 Fuzzy Inference System, 3 inputs are required, 2 of them are the gradients with respect to x-axis and y-axis, calculated with (1), to which we will call DH and DV respectively.

The Sobel edges detector uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

$$Sobel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad Sobel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{1}$$

Where Sobely y Sobelx are the Sobel Operators throughout x-axis and y-axis.

If we define *I* as the source image, $g_x$ and $g_{\mathrm{y}}$ are two images which at each point contain the horizontal and vertical derivative approximations, the latter are computed as (2) and (3).

$$g_x = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{x,i,j} * I_{r+i-2,c+j-2} \tag{2}$$

$$g_y = \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} Sobel_{y,i,j} * I_{r+i-2,c+j-2} \tag{3}$$

Where *gx* and *gy* are the gradients along axis-x and axis-y, and * represents the convolution operator.

The other input is a filter that calculates when applying a mask by convolution to the original image. The low-pass filter hMF (4) allow us to detect image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image.

The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favor false edges by effect of noise.

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{4}$$

Then the inputs for FIS type 2 are: DH=$g_{x,}$ DV=$g_{y,}$ M= *hMF*I,* where * is the convolution operator, and de output is a column vector contains the values of the image edges, and we can represent that in graphics shown in figure 4. The Edge Image is smaller than the original because the result of convolution operation is a central matrix where the convolution has a value. Then in our example, each image with dimension 112x92 is reduced to 108x88.

The inference rules and membership function parameters allow to calculate a gray value between -4.5 and 1.5 for each pixel, where the most negative values corresponds to the dark tone in the edges of the image. Then if we see the rules, only when the increment value of the inputs DH and DV are low the output is HIGH or clear (the background), in the rest of rules the output is LOW or dark (the edges). The complete set of fuzzy rules is given as follows (Castro et al., 2006):

1. If (DH is LOW) and (DV is LOW) then (EDGES is HIGH) (1)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is LOW) (1)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is LOW) (1)
4. If (M is LOW) and (DV is MEDIUM) then (EDGES is LOW) (1)
5. If (M is LOW) and (DH is MEDIUM) then (EDGES is LOW) (1)

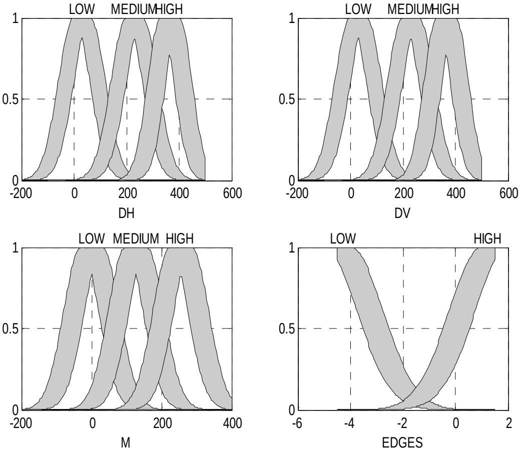


**Fig. 4.** Membership Function for the Type-2 FIS Edge Detector

The edge detector allows us to ignore the background color. We can see in this database of faces, different tones present for the same or another person. Then we eliminate a possible influence of a bad classification by the neural network, without losing detail in the image. Another advantage of edge detector is that the values can be normalized to a homogenous value range, independently the light, contrast or background tone in each image. At the examples in figure 5, all the edges in the images have a

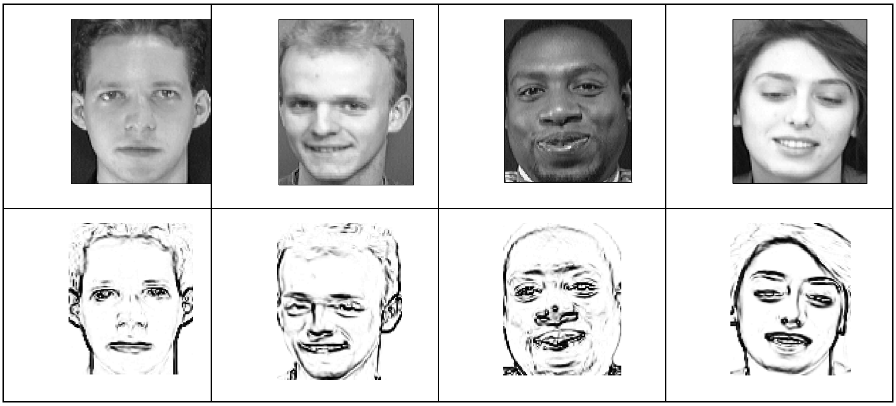


**Fig. 5.** Examples of edge detection with the Type-2 FIS method

minimum value of -3.8 and a maximum value of 0.84. In particular for neural network training, we find these values to make the training faster: the mean of the values is near 0 and the standard deviation is near 1 for all the images.

## 3 The Modular Structure

The design of the Modular Neural Network consists of 3 monolithic feedforward neural networks (Sharkey, 1999), each one trained with a supervised method with the first 7 samples of the 40 images. Then the edges vector column is accumulated until the number of samples to form the input matrix for the neural networks as it is in the scheme of figure 7. Once the complete matrix of images is divided in 3 parts, each module is training with a correspondent part, with some rows of overlap.

The target to the supervised training method consist of one identity matrix for each sample, building one matrix with dimensions 40x(40*number_of_samples).

Each Monolithic Neural Network has the same structure and is trained under the same conditions, like we can see in the next code segment:

```
layer1=200; layer2=200; layer3=number_of_subjects;
net=newff(minmax(p),[layer1,layer2,layer3],{'tansig','tansig','logsig'},'traingdx');
net.trainParam.goal=1e-5;
net.trainParam.epochs=1000;
```

The average number of epochs to meet the goal in each module is of 240, and the required time of 160 seconds.

## 4 Simulation Results

A program was developed in Matlab that simulates each module with the 400 images of the ORL database, building a matrix with the results of the simulation of each module. These matrices are stored in the file “mod.mat” to be analyzed later for the combination of results. We can observe that in the columns corresponding to the training data, the position with a value near one is the image selected correctly. However in the columns that correspond to the test data this doesn’t always happens, reason why it is very important to have a good combination method to recognize more images.

According to exhaustive tests made in the simulation matrices, we know that recognition of the images that were used for the training of the neural networks is of the 100%. Therefore the interest is focused on the recognition of the samples that do not belong to the training set, is to say samples 8,9 and 10. The parameters for the Sugeno Fuzzy Integral that will be inferred will be the Fuzzy Densities, a value between 0 and 1 for each module, which determines the rate for each module. The parameter lambda, according to the theory of fuzzy measures depends on the values of the fuzzy densities, and is calculated by searching for the roots of a polynomial. After the simulation of an image in the Neural Network, the simulation value is the only known parameter to make a decision, then to determine the fuzzy density for each module is the unique available information. For this reason we analyze the values in many simulations

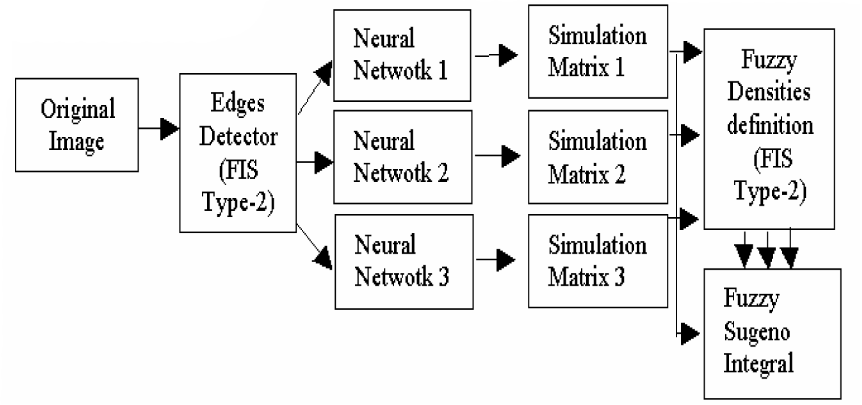


**Fig. 6.** Process of recognition using the type-2 fuzzy modular approach

matrix and decide that each input to the FIS Type-2 corresponds to the maximum value of each column corresponding to the simulation of each module of each one of the 400 images. The process to recognize each one of the images is shown in figure 6.

Then each output corresponds to one fuzzy density, to be applied for each module to perform the fusion of results later with the Fuzzy Sugeno Integral. The inference rules found fuzzy densities near 1 when de maximum value in the simulation is between 0.5 and 1, and near 0 when the maximum value in the simulation is near 0. The fuzzy rules are shown below and membership functions in Figure 7.

1. If (max1 is LOW) then (d1 is LOW) (1)
2. If (max2 is LOW) then (d2 is LOW) (1)
3. If (max3 is LOW) then (d3 is LOW) (1)
4. If (max1 is MEDIUM) then (d1 is HIGH) (1)
5. If (max2 is MEDIUM) then (d2 is HIGH) (1)
6. If (max3 is MEDIUM) then (d3 is HIGH) (1)
7. If (max1 is HIGH) then (d1 is HIGH) (1)
8. If (max2 is HIGH) then (d2 is HIGH) (1)
9. If (max3 is HIGH) then (d3 is HIGH) (1)

Although the rules are very simple, allows to model the fuzziness to rate de modules when the simulation result don’t reach the maximum value 1.

However some of the images don’t reach the sufficient value in the simulation of the three modules, in these cases, do not exists enough information to select an image at the modules combination, and the image is wrongly selected.

In order to measure of objective form the final results, we developed a method of random permutation, which rearranges the samples of each person before the training. Once a permutation is made, the modular neural networks are trained and combined four times to obtain the sufficient information to validate the results. The average recognition rate is of 96.5%.

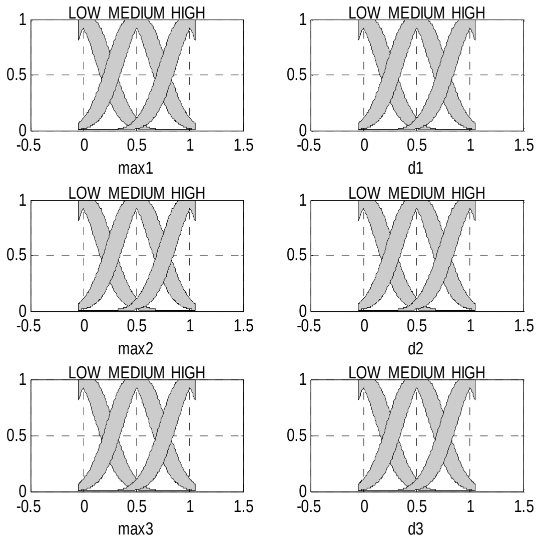


**Fig. 7.** Membership functions for the FIS to find fuzzy densities

We show in Table 1 the summary of simulation results for each of the modules and the average and maximum results of the modular network (after fusion or combination of the results).

**Table 1.** Summary of the Simulation Results with the Hybrid Approach

| Permu-tation | Image Recognition (%) | | | | | |
|---|---|---|---|---|---|---|
| | Train 1 | Train 2 | Train 3 | Train 4 | Average | Maximum |
| 1 | 92.75 | 95 | 92.2 | 93.25 | 93.3 | 95 |
| 2 | 96.5 | 95.25 | 94.25 | 95.5 | 95.375 | 96.5 |
| 3 | 91.5 | 92 | 93.75 | 95.25 | 93.125 | 95.25 |
| 4 | 94.5 | 94.5 | 93.25 | 94 | 94.0625 | 94.5 |
| 5 | 93.75 | 93.5 | 94 | 96 | 94.3125 | 96 |
| | | | | | **94.035** | **96.5** |

## 5 Interval Type-2 Fuzzy Logic for Digital Image Edge Detection

In the area of digital signal processing, methods have been proven to solve the problem of image recognition. Some of them include techniques like binarization, bidimensional filtrate, detection of edges and compression using banks of filters and trees, among others.

Specifically in methods for the detection of edges we can find comparative studies of methods like: Canny, Narwa, Iverson, Bergholm y Rothwell. Others methods can be grouped into two categories: Gradient and Laplacian.

The gradient methods like Roberts, Prewitt and Sobel detect edges, looking for maximum and minimum in first derived from the image. The Laplacian methods like

Marrs-Hildreth do it finding the zeros of second derived from the image (Mendoza and Melin, 2005).

This work is the beginning of an effort for the design of new pre-processing images techniques, using Fuzzy Inference Systems (FIS), which allows feature extraction and construction of input vectors for neural networks with aims of image recognition.

Artificial neural networks are one of the most used techniques in the automatic recognition of patterns, here are some reasons:

- Theoretically any function can be determined.
- Except the input patterns, it is not necessary to provide additional information.
- They are possible to be applied to any type of patterns and to any data type.

The idea to apply artificial neural networks for images recognition, tries to obtain results without providing another data that the original images, of this form the process is more similar to the form in which the biological brain learns to recognize patterns, only knowing experiences of past.

Models with modular neural networks have been designed, that allow recognizing images divided in four or six parts. This is necessary due to the great amount of input data, since an image without processing is of 100x100 pixels, needs a vector 10000 elements, where each one corresponds to pixel with variations of gray tones between 0 and 255 (Mendoza and Melin, 2005).

This chapter shows an efficient Fuzzy Inference System for edges detection, in order to use the output image like input data for modular neural networks. In the proposed technique, it is necessary to apply Sobel operators to the original images, and then use a Fuzzy System to generate the vector of edges that would serve as input data to a neural network.

## 6 Edge Detection by Gradient Magnitude

Although the idea presented in this chapter, is to verify the efficiency of a FIS for edges detection in digital images, from the approaches given by Sobel operator, is necessary to display first results using only the gradient magnitude.

The first image of subject number one of the ORL database will be used as an example (Figure 8). The gray tone of each pixel of this image is a value of between 0 and 255.

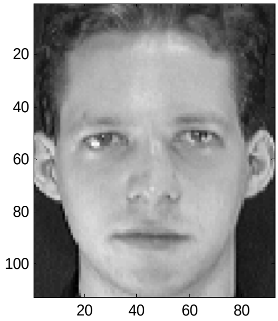


**Fig. 8.** Original Image 1.pgm

In figure 9 the image generated by *gx* is shown, and Figure 10 presents the image generated by *gy*.

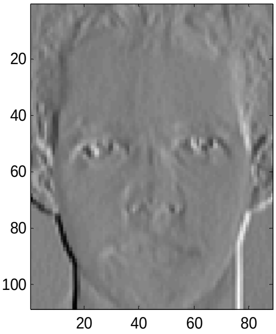


**Fig. 9.** Image given by *gx*

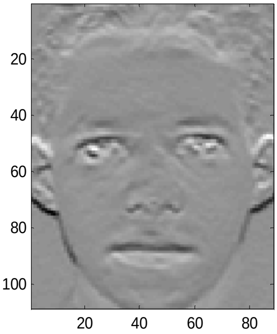


**Fig. 10.** Image given by *gy*

An example of maximum and minimum values of the matrix given by *gx*, *gy* and *g* from the image 1.pgm is shown in Table 2.

**Table 2.** Maximum and Minimum values from 1.pgm, gx, gy and g

| Tone | 1.pgm | gx | gy | g |
|---|---|---|---|---|
| Minimum | 11 | -725 | -778 | 0 |
| Maximum | 234 | 738 | 494 | 792 |

After applying equation (4), g is obtained as it is in Figure 11.

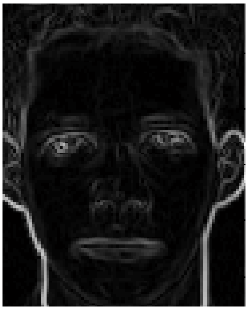


**Fig. 11.** Edges image given by g

## 7 Edge Detection Using Type-1 Fuzzy Logic

A Mamdani FIS was implemented using Type-1 Fuzzy Logic, with four inputs, one output and 7 rules, using the Matlab Fuzzy Logic Toolbox, which is shown in Figure 12.

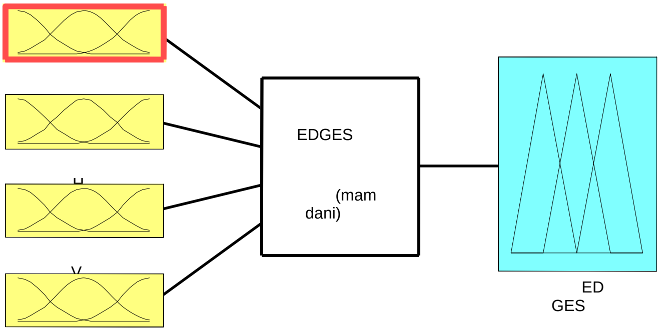


**Fig. 12.** FIS in Matlab Fuzzy Logic Tool Box

For the Type-1Fuzzy Inference System, 4 inputs are required, 2 of them are the gradients with respect to x-axis and y-axis, calculated with equation (2) and equation (3), to which we will call DH and DV respectively.

The other two inputs are filters: A high-pass filter, given by the mask of the equation (5), and a low-pass filter given by the mask of equation (6). The high-pass filter hHP detects the contrast of the image to guarantee the border detection in relative low contrast regions. The low-pass filter hMF allow to detects image pixels belonging to regions of the input were the mean gray level is lower. These regions are proportionally more affected by noise, supposed it is uniformly distributed over the whole image.

The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favor false edges by effect of noise (Miosso and Bauchspiess, 2001).

$$hHP = \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{3}{4} & -\frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \tag{5}$$

$$hMF = \frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{6}$$

Then the inputs for type-1 FIS are:

$DH=g_{x,}$ $DV=g_y$ $HP= hHP*I$ $M= hMF*I$

where * is the convolution operator.

For all the fuzzy variables, the membership functions are of Gaussian type. According to the executed tests, the values in DH and DV, go from -800 to 800, then the ranks in x-axis adjusted as it is in figures 13, 14 and 15, in which the membership functions are:

LOW: gaussmf(43,0),
MEDIUM: gaussmf(43,127),
HIGH: gaussmf(43,255).

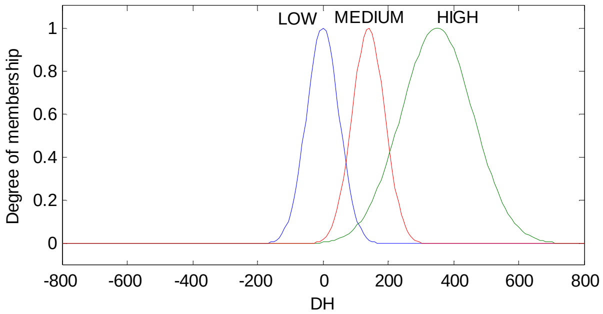


**Fig. 13.** Input variable DH

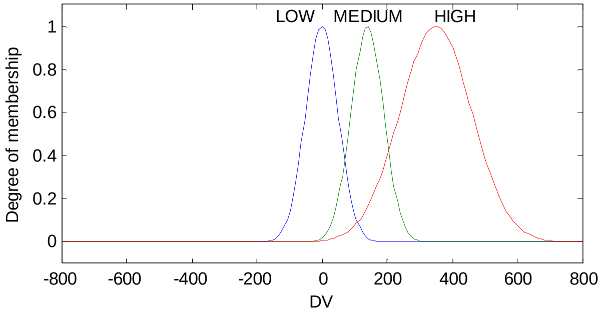


**Fig. 14.** Input variable DV

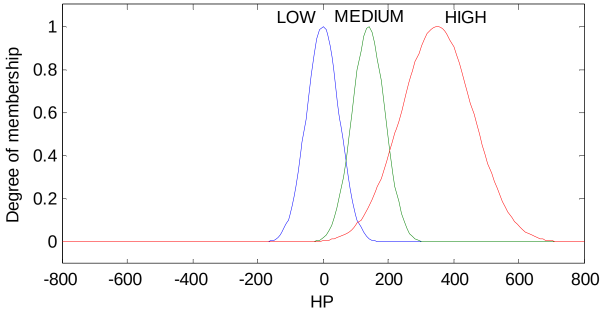


**Fig. 15.** Input variable HP

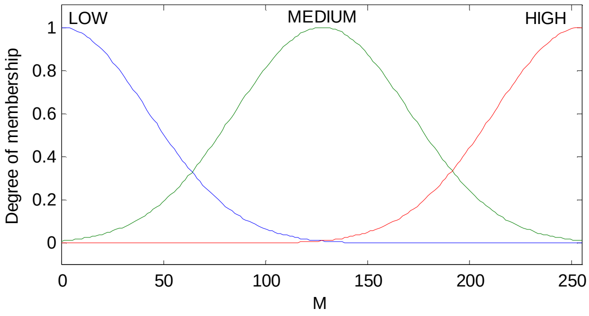


**Fig. 16.** Input variable M

In the case of variable M, the tests threw values in the rank from 0 to 255, and thus the rank in x-axis adjusted, as it is shown in figure 16.

In figure 17 the output variable EDGES is shown, that also adjusted the ranks between 0 and 255, since it is the range of values required to display the edges of an image.

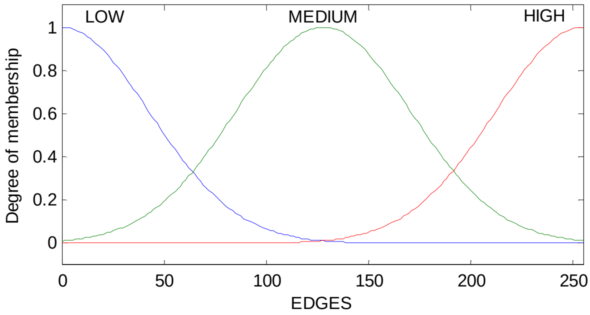


**Fig. 17.** Output variable EDGES

The seven fuzzy rules that allow to evaluate the input variables, so that the exit image displays the edges of the image in color near white (HIGH tone), whereas the background was in tones near black (tone LOW).

1. If (DH is LOW) and (DV is LOW) then (EDGES is LOW)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is HIGH)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is HIGH)
4. If (DH is MEDIUM) and (HP is LOW) then (EDGES is HIGH)
5. If (DV is MEDIUM) and (HP is LOW) then (EDGES is HIGH)
6. If (M is LOW) and (DV is MEDIUM) then (EDGES is LOW)
7. If (M is LOW) and (DH is MEDIUM) then (EDGES is LOW)

The result obtained for image of figure 8 is remarkably better than the one than it was obtained with the method of gradient magnitude, as it is in Figure 18.

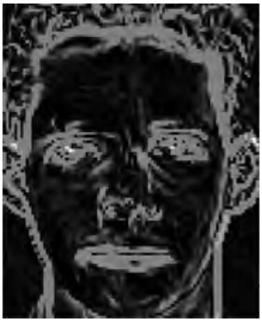


**Fig. 18.** EDGES Image by FIS Type 1

Reviewing the values of each pixel, we see that all fall in the rank from 0 to 255, which is not obtained with the method of gradient magnitude.

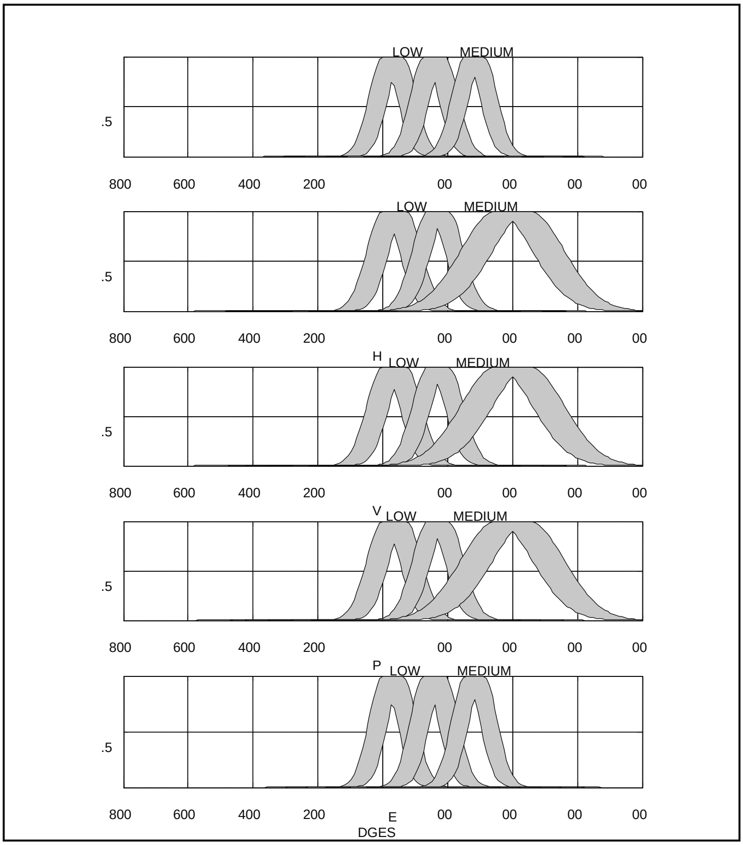


**Fig. 19.** Type-2 fuzzy variables

## 8 Edge Detection Using Type-2 Fuzzy Logic

For the Type-2 FIS, the same method was followed as in Type-1 FIS, indeed to be able to make a comparison of both results. The tests with the type-2 FIS, were executed using the computer program imagen_bordes_fis2.m, which creates a Type-2 Inference System (Mamdani, 1993) by intervals (Mendel, 2001).

The mentioned program creates the type-2 fuzzy variables as it is seen in figure 19. The wide of the FOU chosen for each membership function was the one that had better results after several experiments. The program imagen_bordes_fuzzy2.m was implemented to load the original image, and to apply the filters before mentioned. Because the great amount of data that the fuzzy rules must evaluate, the image was divided in four parts, and the FIS was applied to each one separately. The result of each evaluation gives a vector with tones of gray by each part of the image, in the end is the complete image with the edges (figure 20).

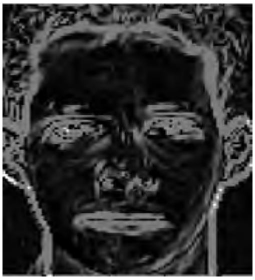


**Fig. 20.** EDGES Image by the Type-2 FIS

## 9 Comparison of Results

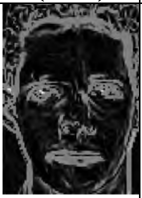
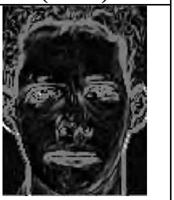
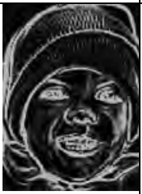
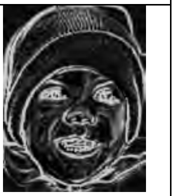
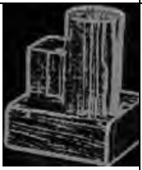
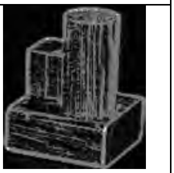
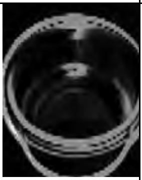
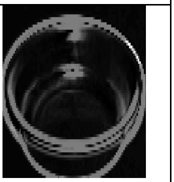
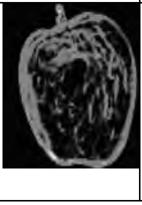
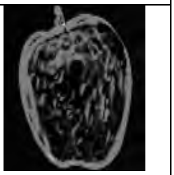
The first results of several tests conducted in different images can be appreciated in table 3.

At first, the results with the Type-1 FIS and Type-2 FIS are seen to be very similar. However thinking about that to show the images with a dark background it could confuse the contrast of tones, tests were done inverting the consequent of the rules, so that the edges take the dark tone and the bottom the clear tone, the rules changed to the following form:

1. If (DH is LOW) and (DV is LOW) then (EDGES is HIGH)
2. If (DH is MEDIUM) and (DV is MEDIUM) then (EDGES is LOW)
3. If (DH is HIGH) and (DV is HIGH) then (EDGES is LOW)
4. If (DH is MEDIUM) and (HP is LOW) then (EDGES is LOW)
5. If (DV is MEDIUM) and (HP is LOW) then (EDGES is LOW)
6. If (M is LOW) and (DV is MEDIUM) then (EDGES is HIGH)
7. If (M is LOW) and (DH is MEDIUM) then (EDGES is HIGH)

Fuzzy Systems were tested both (Type-1 and Type-2), with the new fuzzy rules and same images, obtaining the results that are in Table 4.

**Table 3.** Results of Edge Detection by FIS1 and FIS2 (Dark Background)

| Original Image | EDGES<br>(FIS 1) | EDGES<br>(FIS 2) |
|---|---|---|
| |  |  |
| |  |  |
| |  |  |
| |  |  |
| |  |  |

In this second test can be appreciated a great difference between the results obtained with the FIS 1 and FIS 2, noticing at first a greater contrast in the images obtained with the FIS 1 and giving to the impression of a smaller range of tones of gray in the type-2 FIS.

In order to obtain an objective comparison of the images, histograms were elaborated respectively corresponding to the resulting matrices of edges of the FIS 1 and FIS 2, which are in table 5.

The histograms show in the y-axis the range of tones of gray corresponding to each image and in x-axis the frequency in which he appears pixel with each tone.

**Table 4.** Results of Edge Detection by FIS1 and FIS2 (Clear Background)

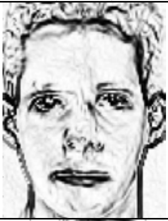
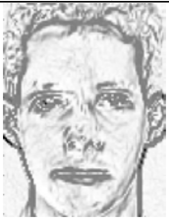
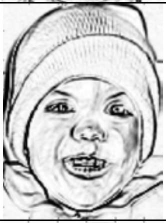
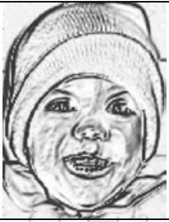
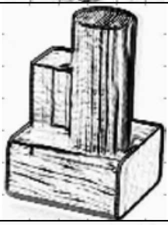
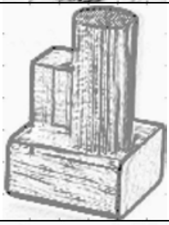
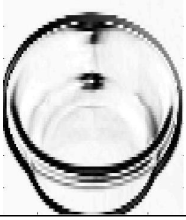
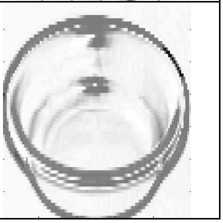
| EDGES<br>(FIS 1) | EDGES<br>(FIS 2) |
|---|---|
|  |  |
|  |  |
|  |  |
|  |  |

**Table 5.** Histograms of the Resulting Images of Edge Detection by the Gradient Magnitude, FIS1 and FIS 2 methods

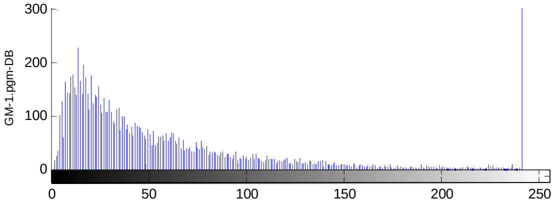
| IMAGE: 1.PGM |
|---|
| METHOD: GRADIENT MAGNITUDE |
|  |
| METHOD: FIS 1 (CLEAR BACKGROUND) |

**Table 5.** (*continued*)

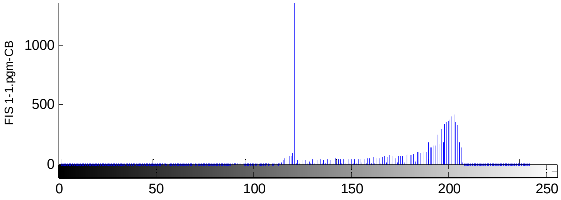
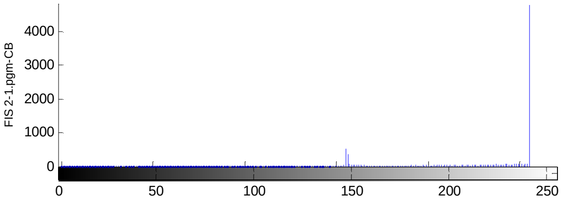
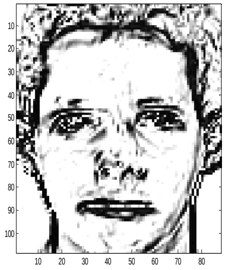
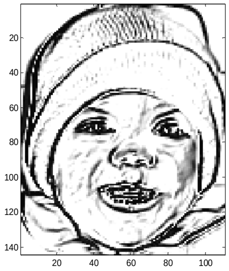
| |
|---|
|  |
| METHOD: FIS 2 (CLEAR BACKGROUND) |
|  |

**Table 6.** Type-2 FIS Edges Images including Pixels with Tones between 150 and 255

| BORDERS IMAGE | DIMENSION (pixels) | PIXELS INCLUDED |
|---|---|---|
|  10<br>20<br>30<br>40<br>50<br>60<br>70<br>80<br>90<br>100<br>10<br>20<br>30<br>40<br>50<br>60<br>70<br>80 | 108x88<br><br>(9504) | 4661<br><br>49 % |
|  20<br>40<br>60<br>80<br>100<br>120<br>140<br>20<br>40<br>60<br>80<br>100 | 144x110<br><br>(15840) | 7077<br><br>44.6 % |

As we can observe, unlike detector FIS1, with FIS2 the edges of an image could be obtained from very complete form, only taking the tones around 150 and 255.

As a last experiment, in this occasion to the resulting images of the Type-2 FIS the every pixel out of the range between 50 and 255 was eliminated.

Table 6 shows the amount of elements that was possible to eliminate in some of the images, we see that the Type-2 Edges Detector FIS allows to using less than half of the original pixels without losing the detail of the images. This feature could be a great advantage if these images are used like input data in neural networks for detection of images instead the original images.

## 10 Systematic Design of a Stable Type-2 Fuzzy Logic Controller

Stability has been one of the central issues concerning fuzzy control since Mamdani's pioneer work (Mamdani and Assilian, 1975). Most of the critical comments to fuzzy control are due to the lack of a general method for its stability analysis.

But as Zadeh often points out, fuzzy control has been accepted by the fact that it is task-oriented control, while conventional control is characterized as setpoint-oriented control, and hence do not need a mathematical analysis of stability. Also, as Sugeno has mentioned, in general, in most industrial applications, the stability of control is not fully guaranteed and the reliability of a control hardware system is considered to be more important than the stability (Sugeno, 1999).

The success of fuzzy control, however, does not imply that we do not need a stability theory for it. Perhaps the main drawback of the lack of stability analysis would be that we cannot take a model-based approach to fuzzy control design. In conventional control theory, a feedback controller can be primarily designed so that a close-loop system becomes stable. This approach of course restricts us to setpoint-oriented control, but stability theory will certainly give us a wider view on the future development of fuzzy control.

Therefore, many researchers have worked to improve the performance of the FLC's and ensure their stability. Li and Gatland in 1995 proposed a more systematic design method for PD and PI-type FLC's. Choi, Kwak and Kim (Choi et al., 2000) present a single-input FLC ensuring stability. Ying in 1994 presented a practical design method for nonlinear fuzzy controllers, and many other researchers have results on the matter of the stability of FLC's, in (Castillo et al., 2005) and (Cázarez et al., 2005) presents an extension of the Margaliot work (Margaliot and G. Langholz, 2000) to built stable type-2 fuzzy logic controllers in Lyapunov sense.

This work is based on Margaliot´s work (Margaliot and Langholtz, 2000), we use the Fuzzy Lyapunov Synthesis to build an Stable Type-2 Fuzzy Logic Controller for a 1 Degree of Freedom (DOF) manipulator robot, first without gravity effect to prove stability, and then with gravity effect to prove the robustness of the controller. The same criteria can be used for any number of DOF manipulator robots, linear or nonlinear, and any kind of plants.

## 11 Fuzzy Logic Controllers

### 11.1 Type-1 Fuzzy Logic Control

Type-1 FLCs are both intuitive and numerical systems that map crisp inputs to a crisp output. Every FLC is associated with a set of rules with meaningful linguistic interpretations, such as

$R^l$ : If $x_1$ is $F_1^l$ and $x_2$ is $F_2^l$ and … and $x_n$ is $F_n^l$ Then $w$ is $G^l$

which can be obtained either from numerical data, or experts familiar with the problem at hand. Based on this kind of statement, actions are combined with rules in an antecedent/consequent format, and then aggregated according to approximate reasoning theory, to produce a nonlinear mapping from input space $U = U_1 x U_2 x ... U_n$ to the output space $W$, where $F_k^l \subset U_k, k = 1,2,...,n$, are the antecedent type-1 membership functions, and $G^l \subset W$ is the consequent type-1 membership function. The input linguistic variables are denoted by $u_k, k = 1,2,...,n$, and the output linguistic variable is denoted by $w$.

A Fuzzy Logic System (FLS), as the kernel of a FLC, consist of four basic elements (Fig. 21): the type-1 fuzzyfier, the fuzzy rule-base, the inference engine, and the type-1 defuzzyfier. The fuzzy rule-base is a collection of rules in the form of $R^l$, which are combined in the inference engine, to produce a fuzzy output. The type-1 fuzzyfier maps the crisp input into type-1 fuzzy sets, which are subsequently used as inputs to the inference engine, whereas the type-1 defuzzyfier maps the type-1 fuzzy sets produced by the inference engine into crisp numbers.

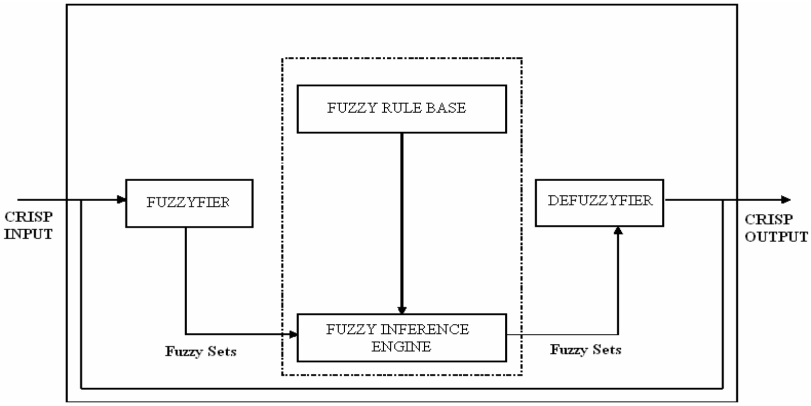


**Fig. 21.** Structure of type-1 fuzzy logic system

Fuzzy sets can be interpreted as membership functions $u_X$ that associate with each element $x$ of the universe of discourse, $U$, a number $u_X(x)$ in the interval [0,1]:

$$u_X : U \to [0,1] \tag{7}$$

For more detail of Type-1 FLS see (Chen and Pham, 2001).

## 11.2 Type-2 Fuzzy Logic Control

As with the type-1 fuzzy set, the concept of type-2 fuzzy set was introduced by Zadeh as an extension of the concept of an ordinary fuzzy set (Zadeh, 1975).

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain. On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact, and measurement uncertainties (Mendel, 2000).

It is known that type-2 fuzzy set let us to model and to minimize the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand that type-1 fuzzy sets; hence, their use is not widespread yet.

Similar to a type-1 FLS, a type-2 FLS includes type-2 fuzzyfier, rule-base, inference engine and substitutes the defuzzifier by the output processor. The output processor includes a type-reducer and a type-2 defuzzyfier; it generates a type-1 fuzzy set output (from the type reducer) or a crisp number (from the defuzzyfier). A type-2 FLS is again characterized by IF-THEN rules, but its antecedent and consequent sets are now of type-2. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades. A model of a type-2 FLS is shown in Figure 22.

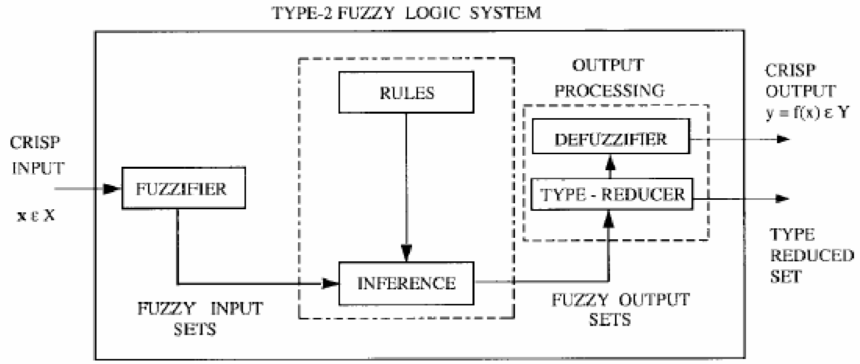


**Fig. 22.** Structure of type-2 fuzzy logic system

In the case of the implementation of type-2 FLCs, we have the same characteristics as in type-1 FLC, but we now use type-2 fuzzy sets as membership functions for the inputs and for the outputs. Fig. 23 shows the structure of a control loop with a FLC.

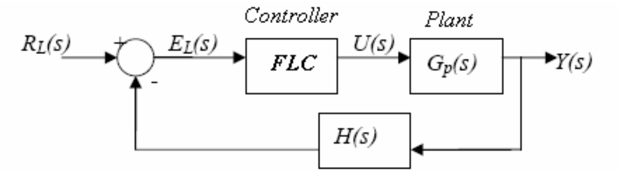


**Fig. 23.** Fuzzy control loop

## 12 Systematic and Design of Stable Fuzzy Controllers

For our description we consider the problem of designing a stabilizing controller for a 1DOF manipulator robot system depicted in Fig. 24. The state-variables are $x_1 = \theta$ - the robot arm angle, and $x_2 = \dot{\theta}$ - its angular velocity. The system's actual dynamical equation, which we will assume that are unknown, is as shown in equation (8) (Paul and Yang, 1999):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{8}$$

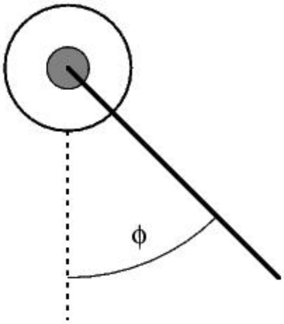


**Fig. 24.** 1DOF Manipulator robot

To apply the fuzzy Lyapunov synthesis method, we assume that the exact equations are unknown and that we have only the following partial knowledge about the plant (see Figure 24):

1. The system may have really two degrees of freedom $\theta$ and $\dot{\theta}$, referred to as $x_1$ and $x_2$, respectively. Hence, $\dot{x}_1 = x_2$.
2. $\dot{x}_2$ is proportional to $u$, that is, when $u$ increases (decreases) $\dot{x}_2$ increases (decreases).

To facilitate our control design we are going to suppose no gravity effect in our model, see (equation 9).

$$ml^2\ddot{q} = \tau \tag{9}$$

Our objective is to design the rule-base of a fuzzy controller that will carry the robot arm to a desired position $x_1 = \theta d$. We choose (10) as our Lyapunov function candidate. Clearly, $V$ is positive-definite.

$$V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) \tag{10}$$

Differentiating $V$, we have equation (11),

$$\dot{V} = x_1\dot{x}_1 + x_2\dot{x}_2 = x_1x_2 + x_2\dot{x}_2 \tag{11}$$

Hence, we require:

$$x_1 x_2 + x_2 \dot{x}_2 < 0 \tag{12}$$

We can now derive sufficient conditions so that condition (12) holds: If $x_1$ and $x_2$ have opposite signs, then $x_1 x_2 < 0$ and (12) will hold if $\dot{x}_2 = 0$; if $x_1$ and $x_2$ are both positive, then (12) will hold if $\dot{x}_2 < -x_1$; and if $x_1$ and $x_2$ are both negative, then (12) will hold if $\dot{x}_2 > -x_1$.

We can translate these conditions into the following fuzzy rules:

- If $x_1$ is *positive* and $x_2$ is *positive* then $\dot{x}_2$ must be *negative big*
- If $x_1$ is *negative* and $x_2$ is *negative* then $\dot{x}_2$ must be *positive big*
- If $x_1$ is *positive* and $x_2$ is *negative* then $\dot{x}_2$ must be *zero*
- If $x_1$ is *negative* and $x_2$ is *positive* then $\dot{x}_2$ must be *zero.*

However, using our knowledge that $\dot{x}_2$ is proportional to $u$, we can replace each $\dot{x}_2$ with $u$ to obtain the fuzzy rule-base for the stabilizing controller:

- If $x_1$ is *positive* and $x_2$ is *positive* Then $u$ must be *negative big*
- If $x_1$ is *negative* and $x_2$ is *negative* Then $u$ must be *positive big*
- If $x_1$ is *positive* and $x_2$ is *negative* Then $u$ must be *zero*
- If $x_1$ is *negative* and $x_2$ is *positive* Then $u$ must be *zero.*

It is interesting to note that the fuzzy partitions for $x_1$, $x_2$, and $u$ follow elegantly from expression (11). Because $\dot{V} = x_2(x_1 + \dot{x}_2)$, and since we require that $\dot{V}$ be negative, it is natural to examine the signs of $x_1$ and $x_2$; hence, the obvious fuzzy partition is *positive, negative*. The partition for $\dot{x}_2$, namely *negative big, zero, positive big* is obtained similarly when we plug the linguistic values *positive, negative* for $x_1$ and $x_2$ in (11). To ensure that $\dot{x}_2 < -x_1$ $(\dot{x}_2 > -x_1)$ is satisfied even though we do not know $x_1$'s exact magnitude, only that it is *positive* (*negative*), we must set $\dot{x}_2$ to *negative big* (*positive big*). Obviously, it is also possible to start with a given, predefined, partition for the variables and then plug each value in the expression for $\dot{V}$ to find the rules. Nevertheless, regardless of what comes first, we see that fuzzy Lyapunov synthesis transforms classical Lyapunov synthesis from the world of exact mathematical quantities to the world of computing with words (Zadeh, 1996).

To complete the controllers design, we must model the linguistic terms in the rule-base using fuzzy membership functions and determine an inference method. Following (Wang, 1997), we characterize the linguistic terms *positive, negative, negative big,* *zero* and *positive big* by the type-1 membership functions shown in Fig. 25 for a Type-1 Fuzzy Logic Controller, and by the type-2 membership functions shown in Figure 26 for a Type-2 Fuzzy Logic Controller. Note that the type-2 membership functions are extended type-1 membership functions.

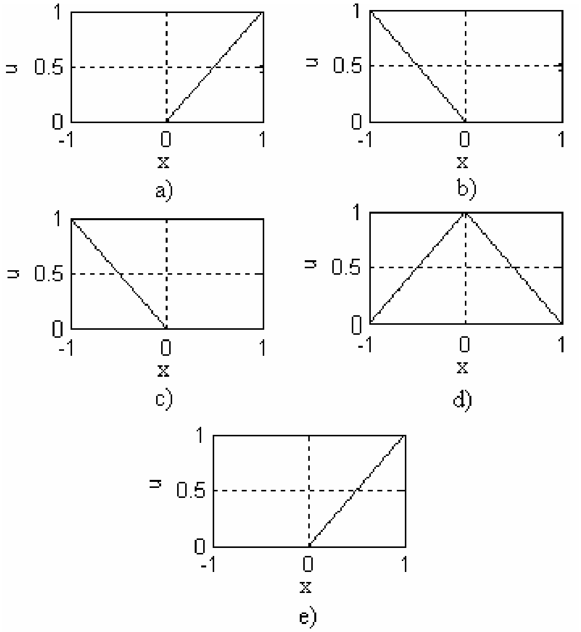


**Fig. 25.** Set of type-1 membership functions: a) positive, b)negative, c) negative big, d) zero and e) positive big

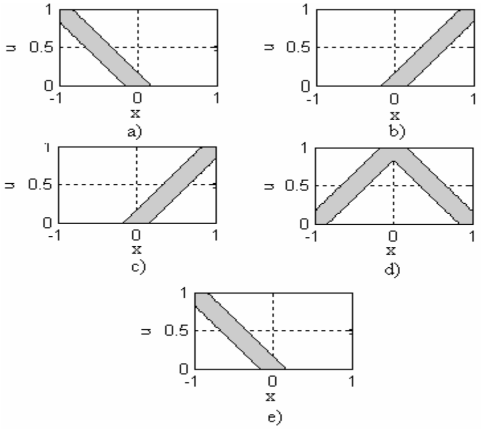


**Fig. 26.** Set of type-2 membership functions: a)negative, b) positive, c) positive big, d) zero and e) negative big

To this end, we had systematically developed a FLC rule-base that follows the Lyapunov Stability criterion. In Section 13 we present some experimental results using our fuzzy rule-base to build a Type-2 Fuzzy Logic Controller.

## 13 Experimental Results

In Section 12 we had systematically developed a stable FLC rule-base, and now we are going to show some experimental results using our stable rule-base to build a Type-2 FLC. The plant description used in the experiments is the same shown in Section 12.

Our experiments were done with Type-1 Fuzzy Sets and Interval Type-2 Fuzzy Sets. In the Type-2 Fuzzy Sets the membership grade of every domain point is a crisp set whose domain is some interval contained in [0,1] (Mendel, 2000). On Fig. 26 we show some Interval Type-2 Fuzzy Sets, and for each fuzzy set, the grey area is known as the Footprint of Uncertainty (FOU) (Mendel, 2000), and this is bounded by an upper and a lower membership function as shown in Fig. 27.

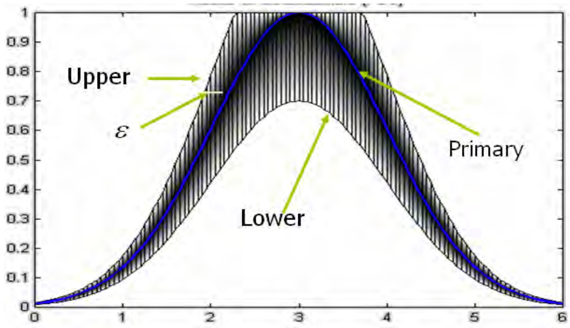


**Fig. 27.** Interval Type-2 Fuzzy Set

In our experiments we increase and decrease the value of $\varepsilon$ to the left and to the right side having a $\varepsilon L$ and a $\varepsilon R$ values respectively to determine how much the FOU can be extended or perturbed without losing stability in the FLC.

We did make simulations with initial conditions of $\theta$ having values in the whole circumference [0, $2\pi$ ], and the desired angle $\theta d$ having values in the same range. The initial conditions considered in the experiments shown in this paper are an angle $\theta = 0rad$ and $\theta_d = 0.1rad$ .

In Fig. 28 we show a simulation of the plant made with a Type-1 FLC, as can be seen, the plant has been regulated in around 8 seconds, and in Fig. 29 we show the graph of equation (11) which is always negative defined and consequently the system is stable.

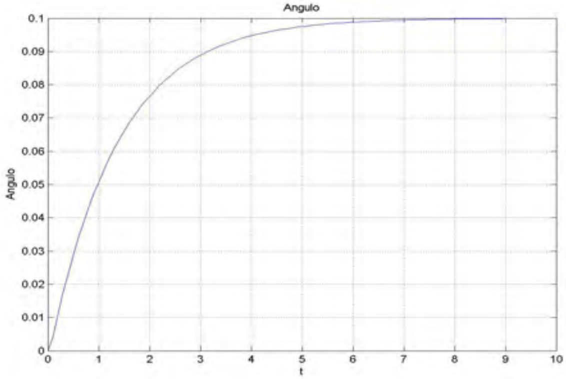


**Fig. 28.** Response for the Type-1 FLC

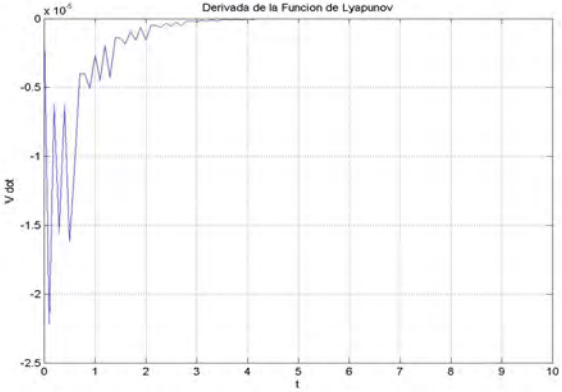


**Fig. 29.** $\dot{V}$ for the Type-1 FLC

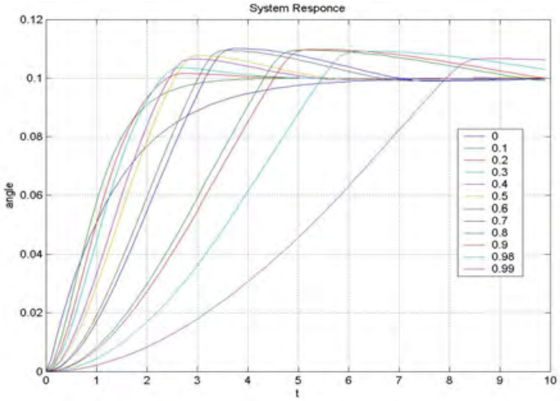


**Fig. 30.** Response for the Type-2 FLC ( $\varepsilon \rightarrow [0,1)$ )

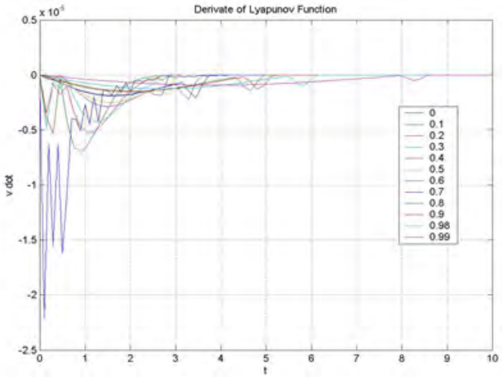


**Fig. 31.** $\dot{V}$ for the Type-2 FLC ( $\varepsilon \longrightarrow [0,1]$ )

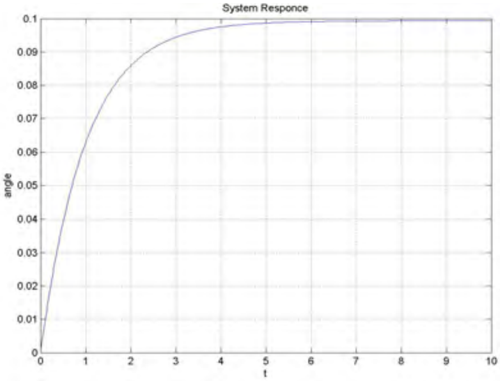


**Fig. 32.** Response for the Type-1 FLC

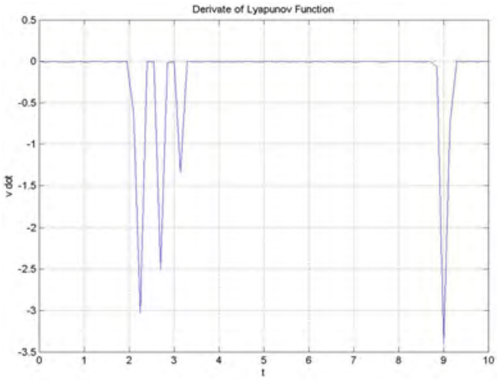


**Fig. 33.** $\dot{V}$ for the Type-1 FLC

Figure 30 shows the simulation results of the plant made with the Type-2 FLC increasing and decreasing $\varepsilon$ in the range of [0,1], and as can be seen the plant has been regulated in around 10 seconds, and the graph of equation (11), which is depicted in Fig. 31 is always negative defined and consequently the system is stable. As we can seen, the time response is increasing when the value of $\varepsilon$ is increasing.

With the variation of $\varepsilon$ in the definition of the FOU, the control surface changes proportional to the change of $\varepsilon$, for this reason, the value of *u* for $\varepsilon \geq 1$ is practically zero, and the plant does not have physical response. To test the robustness of the built Fuzzy Controller, now we are going to use the same controller designed in Section 12, but at this time, we are going to use it to control equation (8) considering the gravity effect as shown in equation (13).

$$ml^2\ddot{q} + gml\cos q = \tau \tag{13}$$

In Figure 32 we can see a simulation of the plant obtained with a Type-1 FLC, and as can be seen, the plant has been regulated in approximately 8 seconds and Figure 33 shows the graph of equation (11) which is always negative defined and consequently the system is stable.

Figure 34 shows the simulation results of the plant obtained with the Type-2 FLC with increasing and decreasing $\varepsilon$ values in the range of [0,1], and the graph of (11) depicted at Fig. 35 is always negative defined and consequently the system is stable. As we can seen, if we use an adaptive gain like in (Castillo et al., 2005) all the cases of $\varepsilon$ can be regulated around 8 seconds.

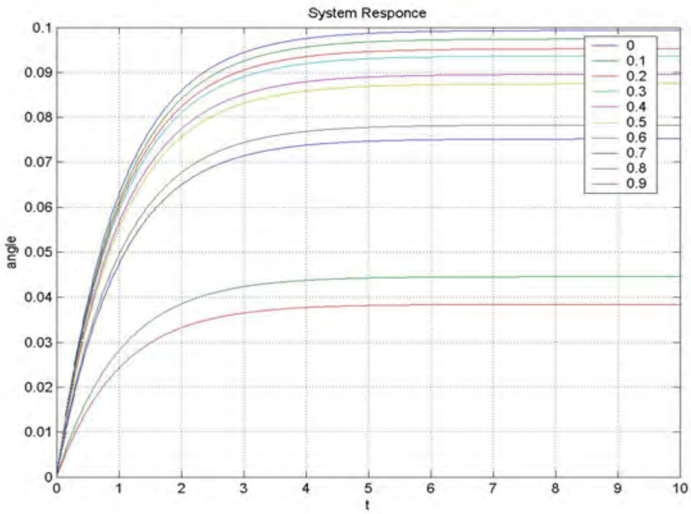


**Fig. 34.** Response for the Type-2 FLC ( $\varepsilon \rightarrow [0,1)$ )

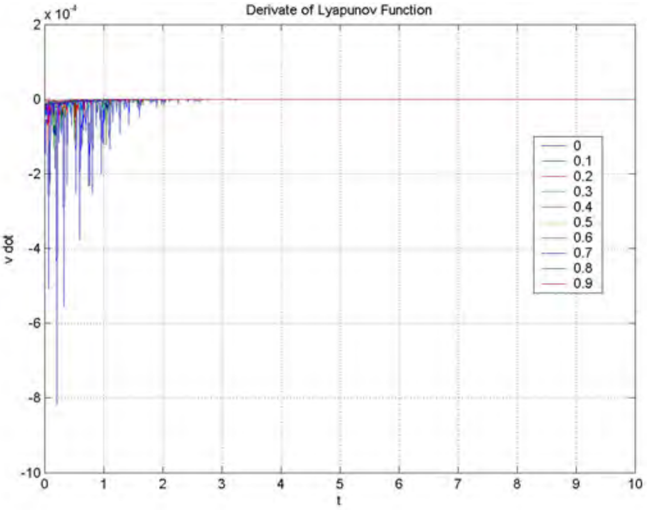


**Fig. 35.** $\dot{V}$ for the Type-2 FLC ( $\varepsilon \rightarrow [0,1]$ )

## 14 Conclusions

In this chapter three applications of interval type-2 fuzzy logic have been described. First, the use of interval type-2 fuzzy logic is used to improve performance on a modular neural network for face recognition. Second, interval type-2 fuzzy logic is used to improve edge detection in image processing. Finally, a method for designing stable interval type-2 fuzzy logic controllers is proposed. In all cases, the results of type-2 fuzzy logic are shown to be superior with respect to the type-1 corresponding ones.

## References

1. Castillo, O., Aguilar, L., Cazarez, N., Rico, D.: Intelligent Control of Dynamical Systems with Type-2 Fuzzy Logic and stability Study. In: Proceedings of the Conference on Artificial Intelligence (ICAI 2005), pp. 400–405 (2005)
2. Castillo, O., Melin, P., Kacprzyk, J., Pedrycz, W.: Hybrid Intelligent Systems: Design and Analysis. Springer, Germany (2007)
3. Castro, J.R., Castillo, O., Martínez-Méndez, L.G.: Tool Box para Lógica Difusa Tipo-2 por Intervalos. In: International Seminar on Computational Intelligence, IEEE - CIS Mexico Chapter, Tijuana, Mexico, October 9-11 (CD-rom proceedings) (2006)
4. Cázarez, N.R., Cárdenas, S., Aguilar, L., Castillo, O.: Lyapunov Stability on Type-2 Fuzzy Logic Control. In: IEEE-CIS International Seminar on Computational Intelligence, México Distrito Federal, México, October 17-18 (Proceedings in CD-rom) (2005)
5. Chen, G., Pham, T.T.: Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems. CRC Press, Boca Raton (2001)
6. Choi, B.J., Kwak, S.W., Kim, B.K.: Design and Stability Analysis of Single-Input Fuzzy Logic Controller. IEEE Trans. Fuzzy Systems 30, 303–309 (2000)

7. Chuang, M.-M., Chang, R.-F., Huang, Y.-L.: Automatic, Facial Feature Extraction in Model-based Coding. Journal of Information Science and Engineering 16, 447–458 (2000)
8. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice-Hall, New York (1995)
9. Kolmanovsky, I., McClamroch., N.H.: Developments in Nonholonomic Nontrol Problems. IEEE Control Syst. Mag. 15, 20–36 (1995)
10. Kosko, B.: Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. Prentice-Hall, Englewood Cliffs (1992)
11. Lee, T.H., Leung, F.H.F., Tam, P.K.S.: Position Control for Wheeled Mobile Robot Using a Fuzzy Controller, pp. 525–528. IEEE, Los Alamitos (1999)
12. Li, H.-X., Gatland, H.B.: A new methodology for designing a fuzzy logic controller. IEEE Trans. Systems Man and Cybernetics 25, 505–512 (1995)
13. Mamdani, E.H., Assilian, S.: An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller. International Journal of Man-Machine Studies 7, 1–13 (1975)
14. Mamdani, E.H.: Twenty years of Fuzzy Control: Experiences Gained and Lessons Learned. In: Proc. 2nd IEEE International Conference on Fuzzy Systems, San Francisco, CA, pp. 339–344 (1993)
15. Margaliot, M., Langholz, G.: New Approaches to Fuzzy Modeling and Control. World Scientific Press, Singapore (2000)
16. Melin, P., Castillo, O.: Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing. Springer, Heidelberg (2005)
17. Melin, P., Mancilla, A., Lopez, M., Mendoza, O.: A hybrid modular neural network architecture with fuzzy Sugeno integration for time series forecasting. Journal of Applied Soft Computing (accepted for publication, 2007)
18. Mendel, J.M.: Uncertainty, fuzzy logic, and signal processing. Signal Processing Journal 80, 913–933 (2000)
19. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, New Jersey (2001)
20. Mendoza, O., Melin, P.: The Fuzzy Sugeno Integral as a Decision Operator in the Recognition of Images with Modular Neural Networks. In: International Conference on Fuzzy Systems, Neural Networks and Genetic Algorithms FNG 2005 (Proceedings in CD-rom) (2005)
21. Mendoza, O., Melin, P.: Sistemas de Inferencia Difusos Tipo-1 y Tipo-2 Aplicados a la Detección de Bordes en Imágenes Digitales. In: International Seminar on Computational Intelligence, IEEE - CIS Mexico Chapter, Tijuana, Mexico, October 9-11, 2006, Tijuana Institute of Technology (CD-rom proceedings) (2006)
22. Mendoza, O., Melin, P.: The Fuzzy Sugeno Integral as a Decision Operator in the Recognition of Images with Modular Neural Networks. In: Edited Book: Hybrid Intelligent Systems: Design and Analysis, pp. 299–310. Springer, Heidelberg (2007)
23. Sharkey, A.: Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer, London (1999)
24. Sugeno, M.: On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents. IEEE Trans. Fuzzy Systems 7(2) (1999)
25. Wang, L.-X.: A Course in Fuzzy Systems and Control. Prentice-Hall, Englewood Cliffs (1997)
26. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning–1. Journal of Information Sciences 8, 199–249 (1975)
27. Zadeh, L.A.: Fuzzy Logic = Computing with Words. IEEE Transactions on Fuzzy Systems 4(2) (1996)

# Processing of Information Microgranules within an Individual's Society

Bruno Apolloni[1], Simone Bassis[1], and Antonio G. Zippo[2]

[1] Department of Computer Science, University of Milan, Via Comelico 39/41 20135 Milano, Italy
[2] Department of Mathematics, University of Milan, Via Saldini 50 20133 Milano, Italy

**Abstract.** Consider a web society comprised by a huge number of agents communicating with one another. Each agent enjoys a wide facility in receiving and sending data and a very small capacity for processing them. Hence data constitute a heap of micrgranules of information. Each individual processes data by him- or herself with the sole goal of increasing personal utility. In this sense s/he adapts his/her parameters in a monotone way so as to fit granules. The sole social interaction with others all around aims at maintaining a healthy homeostasis. This goal is achieved through an aging mechanism that is set at the basis of a human-centric policy for producing a dynamic formation of clusters of healthy agents within a population. As a result, agents are specialized in a user-dependent task common to all individuals of a same cluster, and possibly belonging to more than one cluster. We may interpret the process as a dynamic assignment of agent membership degrees to the various clusters: each cluster is ranked with an overall quality index; each agent partitions an overall membership on the single clusters.

## 1 Introduction

Human-centric policy seeks a new brand of the Information Communication Technology (ICT) society that calls for a further step in the direction of an alliance between the axiomatic world of mathematics and the pragmatic world of humans [1]. On the path toward a concrete use of mathematical tools in an everyday operational context, the use of computers marked an enormous advancement in searching for results whose features and existence may be foreseen by the theory but whose actual achievement emerges from long iterations of symbolic or numerical routines. Thus, abandoning a utopian world where each phenomenon is ruled by universal laws deduced from a few founding axioms, scientists moved to another ideal world where everything that is needed may be computed automatically by picking universal routines from a well furnished library [2]. The typical goal concerning a routine from this library is to show that it works for any input without any user intervention (*weaning* [3]) apart from a correct feed of the input. Modern researchers claim that this goal makes sense in relatively limited ranges of input [4]. But very few renounce the idea that the routine works all by itself, simply pushing a computer button, and in a feasible amount of time [5]. While up until the 1980s, theoretical computer scientists considered speaking of heuristics a nice shortage, today user-dependent solutions are viewed in the same negative manner. Nevertheless, since the 1990s some methodologists have launched human-centric features for principled automatic activities such as business optimization

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 233–264.
springerlink.com

[6,7], computer aided diagnosis [8,9], privacy and network security [10,11], web and e-learning [12-14], programming language interfaces [15], industrial automation [16], and so on. In some sense we may consider human-centric computations as an evolution of the expert systems [17] almost definitely dismissed at the end of the past century. Of that paradigm, the new computations share the focus on the information that humans already possess and may reverse on the solution of the problem at hand. Indeed we commonly speak of human-centric information processing, like in the title of this book. The novelty of the new paradigm may be summed up in the two keywords it is usually accompanied by: granular information and social computing.

### 1.1 Tailoring Computations to Humans

Social computing [18] has the *emergence* of a functionality [19] as the final result of the computation. You have a certain number of processing elements (PEs), each devoted to its own task and implicitly coordinated through a dense communication grid, the output composition of which results in a functionality suitable to the coordinator. From a computer science perspective, this is the answer to the general failure of the parallel computation paradigms tossed on supercomputers in the 1980s. Since the cost of synchronizing the thousands of CPU proved unbearable – with the consequence of having in the best case a throughput of a few units on a general-purpose machine [20,21] – we opted for a distributed computation paradigm [22]. This relieves the coordination overheads, since each processor runs on the input broadcast to it by the other processors as soon as they produce it. At the same time, this necessarily reduces the processors computational power, since you cannot expect sophisticated computation on unplanned inputs. We speak indeed of elementary processors. From a social viewpoint functionality, emergence represents a modern take on Taylorism that affects the genotypical rather than phenotypical aspect of human activity. The most common example is the worldwide web. You have a communication grid that, jointly with other massive communication tools such as TV and mobile phones, feeds the user with an avalanche of messages. There are so many of these that the user may process them with very simple means, very often coming to some threshold operation replying “yes” after a saturation level has been reached [23]. This asserts an intriguing saddle point between macro and microcosm, as an analogous information processing is assumed to be at the basis of our body’s intelligent cells (e.g. neurons and T-cells). However, such a connection is based on a different kind of fitness concept as the Hamiltonian of this massive process refers not to a single element of the evolving population but to the whole population, in a close analogy with the idea of custom tailoring.

Information features are equally particular. A PE may process only small bits of information. Hence, even in the case where the global settlement of the computation would be fed with a complete and perfectly clean set of data (a situation so ideal as to be generally unfeasible), the single PE will only handle granules of information. They are partial data that it tries to connect to the surroundings through a limited repertoire of instruments like membership functions, probability models or deduction rules it has available [24]. Seen from a bottom-up perspective (data on bottom, concepts on top), granularity stands for a smearing of numerical data into blobs around them. This calls for specific methods, such as clustering, merging, fuzzy processing and so on, to process these entities [24,25]. The feature is not bounded to a given scale. We may state similarity or fuzzy relations in general between observations of a phenomenon, yet also on models describing the phenomenon, for instance in terms of *behavior distribution* [26] or collaborative fuzzy clustering [1,27], etc. The cooperation request, and social computing as its implementation, comes from the definition of information granularity. We may relieve noise and elicit similarities and decision rules exactly because we confront results coming from the individual granules. For short, with traditional statistics we have a central, possibly sophisticated, processing of the data – for instance to compute their mean or variance [28]. With granular computing we have a possibly disingenuous general-purpose resume of the computations done on the single granules [14].

Along with a small local database, the repertoire of agent instruments represents the core of the person in the role of PE. Actually s/he could run more complex computations on terabyte-sized databases and he really does so in many situations. But when the goal of the computation is highly complex, the best s/he does is organize an army of clones reflecting his/her actually limited conceptual tools w.r.t. the problem and runs them as explained before expecting the mentioned emergence of a solution, for instance according to the general paradigm of learning by ensemble [29]. Such is the situation at the level of individual person. But this scheme may reflect the organization of real communities of living organisms, ranging from single cells to internet virtual community members. In these cases we face (not organize) ensembles of PEs and wonder about their elementary computations, the control rules making them a community in place of random particles, and the global functionality they implement. In this chapter we discuss these items with the aim of showing that even more elementary computations than those generally acknowledged to the PEs are sufficient for obtaining basilar function emergence like homeostasis [30]. It should allow both a more realistic and suitable analysis of biological phenomena and a possible inauspicious forecasting on social phenomena as well [31]. The key ingredient collating the individual’s computations is an age factor singularly affecting PEs under a stabilizing effect of surrounding PEs [32]. Age, as an asymmetry in the time axis, denotes a memory in the processor behavior that in turn is at the basis of some intentionality of their computations [33]. This looks to us as the main distinguishing features between ensembles of inanimate particles like a gas powered by Brownian motion [34] and our processors, which therefore play henceforth the role of (real or artificial) agents [35]. With these subjects we abandon the framework of memoryless Markov processes [36] and move to more complex random processes that we study, for the moment at a purely descriptive level, within the paradigm of $\pi$-calculus [37]. In turn, it is not surprising that this computational paradigm is mostly diffused today as a tool for analyzing complex dynamical systems, again as a distributed (agent-based) procedure to compute system evolutions ruled by extended forms of ordinary differential equations [38]. With this and similar paradigms, such as biological computers [39], we aim at filling the gap between systems hardware and software. In principle we may synthesize in an agent both the elementary computation and the elementary hardware, maybe a body cell, implementing it.

### 1.2 A Specific Agent-Centric Processing System

As for the essentials of our discourse, we claim that the set point tracking hypothesized to be at the basis of many agents' activity is too complex, whereas a monotone behavior of computations paired with an aging mechanism is more realistic and efficient as well. Moreover, in place of the optimization of a possible hard *static* function, like with other social computing paradigms such as ant colonies [40] or swarm computing [41], our goal is just the sum of the benefits produced by the agents over the span of time in a hostile environment. This turns into the goal of maintaining agents alive and healthy for a long time – the mentioned homeostasis. Hence we will study the evolution of the population health as a function of the rule with which the single agents balance health and productivity needs. The mentioned aging factor acts as a stabilizer of an intrinsically nonlinear unstable process [42]. As for age, it has a monotone behavior contrasted by two phenomena: a drastic one causing the death of the individual and a more continuative one carried out by the rest of the population in terms of an equalizing effect that stops the age growth when it notably exceeds the mean age of the population. The former primes an open loop with its own rules whose effects are countered by the birth of new individuals. The latter is a closed loop where the feedback on the single individual comes from the social laws of the entire community. With the addition of a topological structure between agents we prime a clustering process that evolves through a spatial wandering of the cluster contours within the agents' topological space. As a result, we have a set of clusters which the single agent belongs to with a membership degree represented by the aging factor $\gamma$ (see next section). Along the cluster trajectories we will identify stable clusters whose quality is given by the time integral of the sum of activity of the agents significantly belonging to the cluster. The operational meaning of these clusters and their exploitation will depend on the particular context we are in, as will be discussed with the help of a couple of examples.

We organize the chapter as follows. In the next section we describe the agents' life in mathematical terms. Then in Section 3 we show analogies with microorganisms or cell systems such as immune systems and neural networks, drawing hints from their evolution. In Section 4 we identify the simulation tool in terms of $\pi$-calculus in order to favor the communication aspects. Using this tool, we reproduce a template of social phenomena in Section 5. To draw conclusions, in the final section we will discuss the operational value of these simulations and possible scenarios where they may be framed.

## 2 Life of a New Taylored Worker

Life is a struggle and our worker $g$ fights it everyday in his workplace. It is neither a sophisticated nor an extremely explicit battle. He is urged both by his boss and by a general political message to spend his efforts to produce. Here we assume production to be a non highly demanding elaboration of either material or immaterial items provided by surrounding workers ruled by some monotony with the global input received, call it $a$, and a thresholding with its own parameter $\gamma$ so that if $a \geq \gamma$ then $g$ produces in turn an item with great probability. Work is tiring – as well known – and consumes $g$'s energies. Thus $g$ tends to reduce his productivity, say increase $\gamma$ at each production, in order to safeguard the years he has left. Scaling $\gamma$ between 0 and 1, we may say that $\gamma$ is the normalized *age* of $g$, called *strength* elsewhere from an operational perspective. Hence, not far from what has been hypothesized for many microorganisms (e.g. neurons and bacteria), the main rules of $g$, indexed by $i$ within a population of $n$ workers, are the following:

$$\mathrm{P}(S_i = 1) = a_i(s_1, \ldots, s_n) - \gamma_i \tag{1}$$

$$\gamma_i = \gamma_i + \delta s_i \tag{2}$$

where $s_i$ denotes the production index of $g_i$ ($s_i = 1$ if $g_i$ produces an item, 0 otherwise) and $\gamma_i$ increases its value by $\delta$ each time $g_i$ produces.

Life, as usual, is more complex. On the one hand $g_i$ lives into a community of workers. Thus, he can refrain from producing, but not too much in comparison with his mates. It means that he will increase $\gamma_i$ only if his threshold is comparable (not too higher) than the mean threshold of the surrounding community. This denotes a neighboring notion and complicates (2) as follows

$$\gamma_i = \gamma_i + \delta s_i \quad \text{if } \gamma_i < \tau \bar{\gamma} \tag{3}$$

where $\bar{\gamma}$ is the average of the surrounding workers' $\gamma$s and $\tau$ is a suitable constant.

Moreover, worklife is short. Apart from bodily injuries, a worker may retire for a lot of reasons, including competing work offers or simply job abandonment. This is well known to the manager, who promotes knowledge transfer to new entries with a given probability $\rho$. Thus, within the same surroundings, $g_i$ sparks action w.r.t. a new worker $g_j$ is ruled by:

$$\mathrm{P}(L(g_j) = 1) = \rho\alpha(t) \tag{4}$$

where $L(g')=1$ means the hiring of worker $g'$, 0 his latency. $\rho$ is an awaking constant and $\alpha(t)$ a time modulation.

## 2.1 A Simplified Mathematical Model

A first way of synthesizing the dynamics of the population is analytical in terms of a predator-pray PID system [43-45]. To this aim we introduce some simplifications.

Namely, we assume that $g$ hires another worker $g'$ each time he produces an item (fires) and this happens with probability $1 - \gamma$ (as a shortcut of (1)). Hence with probability $\gamma$ he takes a rest. Conversely, a worker may cease his activity through death. We assume the dying probability to be greater when the worker fires, say equal to $\mu_S$, than in the second case, say $\mu_E$. To close the loop we may think the death as a conflict with noxious environment agents – say stress, better offers, and alike – so that if he does not die he possibly, with probability $\mu_R$, destroys one of these noxious agents. Thus, starting with $N$ workers and $R{\cdot}L$ noxious agents (i.e. a population of $L$ environment agents that prove noxious at a rate $R$), the game at population level evolves through the following equations.

$$\begin{cases} N'(t) = N(t)((1-\gamma(t))(1-\mu_S R(t)) - \gamma(t)\mu_E R(t)) \\ R'(t) = -\dfrac{1}{L} N(t)(\gamma(t)(1-\mu_E)\mu_R R(t)) \\ \gamma'(t) = \alpha(1-\gamma(t)) \end{cases} \tag{5}$$

with $\alpha$ a suitable constant to gain $\gamma(t)$ a smooth profile, and disregarding some approximation problems connected for instance with the expectations of product of variables brute force computed as the product of the single variable expectations.

Playing this simplified system we realize that:

a. for fixed concentration of noxious agents and strength ($R' = 0$, $\gamma' = 0$) we come to a simple predator-prey process admitting equilibrium for $N(t)$ when $\gamma = \gamma^* = 1 + \frac{\rho\mu_E}{-1-\rho(\mu_S - \mu_E)}$ (or equivalently when $R = R^* = \frac{1-\gamma}{\mu_S - \gamma(\mu_S - \mu_E)}$). Obviously $N(t)$ moves to an exponential increase as soon as $\gamma < \gamma^*$ (or $R < R^*$), or decrease when $\gamma > \gamma^*$ (or $R > R^*$);
b. for decreasing concentration of noxious agents and fixed strength ($R' \leq 0$, $\gamma' = 0$) we have equilibrium only for trivial conditions ($R = 0$, $\gamma = 1$). Otherwise we have either: i) an exponential decrease of $N(t)$ (in case $\gamma$ is too high to contrast the decimation operated by noxious agents); ii) an exponential increase; or iii) an initial decrease followed by an exponential increase. The latter tracks an initial prevailing of noxious agents contrasting the reproductive ability of the workers, followed by a behavior's inversion due to an unavoidable decrease of the noxious agents density, on one side, and a beneficial increment of $N'(t)$ that essentially preserves survived workers and their offsprings. See Figure 1 (a-c);
c. $\gamma_B$ smoothly variable from a certain value to 1 in (5) has the benefit of greatly slackening and smoothing the dynamics of the population, thus avoiding the explosion of $N(t)$ – an undesirable condition both on the part of the workers and on the part of their boss – on the contrary, giving the population slow decreasing trends (see Figure 1(d-f)) in a wider range of the free parameters than in the case of $\gamma$ fixed. This happens both for parameters denoting the environmental conditions ($R$) and for those characterizing the worker resistance to noxious agents.

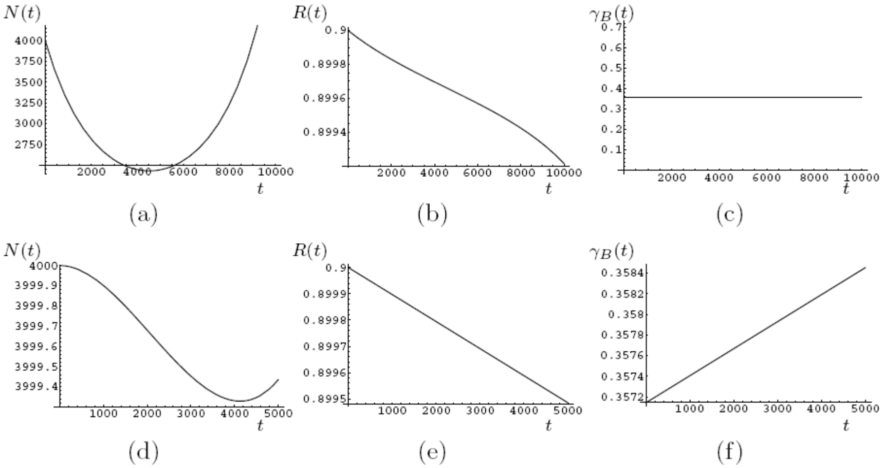


**Fig. 1.** Course of $N(t)$, $R(t)$ and $\gamma_B(t)$ according to (5) in two dynamic's conditions. (a-c) $\gamma_B(t)$ is fixed to 0.36; (d-f) $\gamma_B(t)$ varying with $\alpha$ set to $4{\cdot}10^{-7}$. $L = 10^{10}$, $\mu_E = 0.2$, $\mu_S = 1$ and $\mu_R = 1$.

Following the histories of the single workers, starting from a random distribution of their strengths, we will attain an even more stable behavior of the population as a consequence of an indirect feedback. Actually our population is constituted by individuals working in an open loop, i.e. in the absence of a set point and feed-back for attaining it. On the contrary the adaptation of the individual's attitude is monotone in the direction of increasing $\gamma$ – in this sense we speak of an aging phenomenon – hence without any control action reducing its value. Nevertheless strategy (3) states a closed loop inducing metastable equilibria, before the occurrence of a disruptive event that either precipitates the population size to 0, or explodes it. We may read it as a quorum sensing mechanism well known in natural fields [46], as a suitable fragmentation of the push-pull phenomenon highlighted in Figure 1(a).

### 2.2 A Decision Theory Perspective

From a logical standpoint, we have an additional rationale to asses this paradigm of social computing coming from game theory.

Consider the following *game against nature* [47]. Bob and Alice (B and A for short) are playing the game described below, based on a three-state monotone competition.

**Game 1.** *The game consists in a series of contrasts between B and A on random instances* **s** *drawn from a set* $\mathcal{S}$ *(huge, but finite and discrete) with the following properties:*

- *on each* **s** *B may i) win, ii) lose, or iii) tie with respect to A;*
- *each player owns an integer parameter, let us call them* strength $\gamma_B$ *and* $\gamma_A$ *respectively, which can assume a finite number of values. According to the order relation defeat < tie < victory, for a fixed value of* $\gamma_A$ *the increment of B's strength does not diminish his contest results. Both for minimum* $\gamma_A$ *and whatever* $\gamma_B$*, and for maximum* $\gamma_B$ *and whatever* $\gamma_A$ *B always either wins or ties*
- *the two players have different roles:*
  - *A maintains her strength at a fixed point;*
  - *B increases his strength by one unit whenever he loses;*
- *B's goal is to find (learn) the minimal strength that will let him lose in the future with a probability below a small threshold. His strategy is to achieve this with a good confidence level. That means he will keep playing the game until he achieves this confidence.*

We solved this game elsewhere [48-50] using two run-time statistics: the number $k$ of defeats Bob suffered along the game history and the number $\tilde{k}$ of defeats he should have to bear if he would have played the same contests with the current strength he has.

The main theorem is the following:

**Theorem 1.** *[48] With reference to Game 1 and related statistics k,* $\tilde{k}$ *(realizations of the random variables K,* $\tilde{K}$*), denoting by*

- $U_K$ *the random variable measuring the probability of defeat of a contests' history having k,* $\tilde{k}$ *as statistics of the past contests and continuing with an unlimited sequence of random drawn instances* **s***,*

- *m the number of played steps of our game,*
- *$\mu_\varepsilon$ the Binomial variable of parameters m and $\varepsilon$, and $\upsilon_\varepsilon$ the Binomial variable of parameters k and $\varepsilon$,*

*we have,*

$$\mathrm{P}(\mu_\varepsilon \geq \tilde{k}) \geq \mathrm{P}(U_k \leq \varepsilon) \tag{6}$$

$$\mathrm{P}(U_k \leq \varepsilon) \geq \mathrm{P}(\upsilon_\varepsilon \geq \tilde{k}+1) \tag{7}$$

The great value of this theorem stands in the fact that its claim extends to any game where a monotone effort-effect mechanism characterizes the players, *even in the impossibility of giving it a quantitative description*. A template is represented by the following:

**Game 2.** *Consider the atomic partition of the unitary mass of probability as in the first row of Figure 2, where the balls, in number of $\ell$, own probability $1/\ell$ each. Starting the game, the first $\rho(0)$ balls are white while the remaining ones are gray, and a cursor separates the two sets of balls. At each run t we draw a ball uniformly and replace it after having marked it: if the ball is white we do nothing more, if it is gray we move the cursor $\Delta\rho(t)$ balls right, changing color to the shifted balls (i.e. the new balls to the left of the cursor become white). $\Delta\rho(t)$ is not fixed a priori and can change from iteration to iteration.*

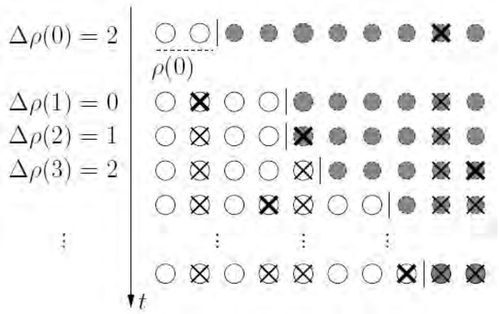


**Fig. 2.** An instance of Game 2 having $\ell = 10$ and $\rho(0) = 2$. The vertical axis reports the iteration number, each labeled with the corresponding $\Delta\rho(t)$. Marked balls are denoted by a cross, while bold crosses denote the balls drawn at each iteration. The cursor position is understood before the ball extraction and eventual updating of it.

In this respect, $k$ represents the number of cursor shifts, $\tilde{k}$ the number of gray marked balls, and losing probability $U_k$ the number of balls on the right of the cursor. A nice way of visualizing (6) and (7) is the seal diagram in Figure 3 showing on the $z$-axis the extremes of 0.90 confidence interval for $U_k$ (namely the 0.05 and 0.95 quantiles of this variable) as a function of $k$ ($x$-axis) and $\tilde{k}$ ($y$-axis). The graph shows that we may estimate the losing probability to have low values with a good accuracy when

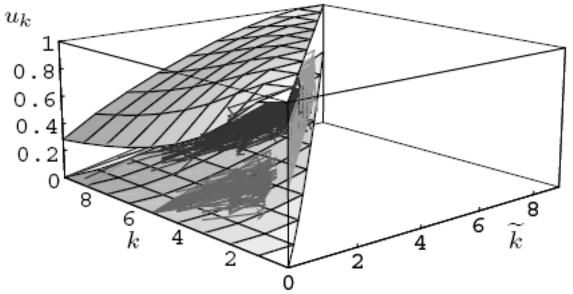


**Fig. 3.** 0.90 confidence intervals for losing probability $u_k$. Course of the lower and upper bounds ($z$-axis) with statistics of $k$ ($x$-axis) and $\tilde{k}$ ($y$-axis).

$k$ is sensibly greater than $\tilde{k}$, hence we may decide stopping increasing $\gamma_B$ when this condition occurs.

A way of enhancing $k-\tilde{k}$ difference is the following. We play two games simultaneously: an *effective game*, where Bob uses the current strength to compute $\tilde{k}$, and a *dummy game*, featuring a strength increment equal to zero, which lets him accumulate $k$. These statistics are the same as in a *virtual game* where a virtual strength maintains its value to a ground value $\gamma_B = 0$, jumping to Bob's current strength $\gamma_B$ exactly when we compute the confidence interval. This game still complies with Game 2 definition and exploits the theoretical results obtained so far. Moreover it numerically denotes that the proposed strategy pushes the great part of the trajectories toward the ($k$ = max, $\tilde{k}=0$) corner. In essence we obtain a better appreciation of the confidence intervals for the same learning process thanks to a better exploitation of the information available on them.

We may identify the worker lever regulating the saving/enrolling attitude with the strength $\gamma_B$ of Bob, as the worker manages strength's increase with an analogous monotone rule. Approximating $\gamma_B$ with $\bar{\gamma}$, B gets with rule (3) the same benefits of the above virtual game. This is the sole hint we obtain from the mathematics of the monotone game. It is not a completely disregardable contribution, however, giving a rationale to natural processes that are so much elementary and universal as much are still obscure.

## 3 The Healthy Homeostasis Problem

Homeostasis is a crucial problem in any living society. It concerns the ability to maintain the system alive and stable in spite of external solicitation, and in compliance with the functionalities the system is requested to carry out. At macro level it concerns ecological systems, such as environment in relation with human activities, or ethological systems, in relation with the equilibria that various animal species may reach in a territory. Going down the scale of phenomena, we discover analogous goals with ever smaller particles, microorganisms or simple cells inside our body. We may get a good look at the problem focusing on brain neurons. The firing mechanism of the neurons may be resumed in many cases without great oversimplification by the heaviside function characterizing the activation of a binary threshold neuron. Namely:

*a neuron fires if it is sufficiently stimulated by the surrounding neurons.*

Rather, in an artificial neural network, it is the timing that is generally oversimplified. In the latter framework we prefer working with a step function rather than a spike function, generally disregarding firing dynamics features such as latency time, etc. The main features coincide however. In both cases wires/synapses transmit a weighted signal from one neuron to another, and the latter fires when the weighted sum of the incoming signals trespasses a given threshold. Hiding complications coming from the temporal window over which summation is done (e.g. possible smoothing of the signal with time and so on) is a very simple mechanism, as primitive as the buildup of stress that makes our patience explode. We may actually prove that such a neural network may compute any computable function [51], where the output of the computation depends on the value and sign (positive/negative) of the weights affecting the incoming signals (connection weights) and other parameters of the networks such as its layout, thresholds, etc.

Things work differently for the plasticity of the neuron parameters. The Hebbian rule, which is the most accredited *learning rule* in the biologists' community [52], is never employed for artificial intelligence purposes, even in the case of unsupervised learning, because of its instability. The typical pattern is that the network tends to strengthen its ability to reproduce some patterns (those already learned) at the expense of the remaining ones, which it totally ignores. Thus the additional terms to the connection weight (+Δ if the states of the neurons at its extremes coincide, –Δ elsewhere) is molded through multiplicative coefficients and regularization terms aimed at avoiding weight divergence. These improvements have however the great drawback of losing the asynchrony of the neuron's lives. They require either a space coordination such as in the backpropation family of algorithms [22] or a time coordination such as in Boltzmann machines [53].

To maintain the independence of the lives of each neuron, biologists prefer to introduce an additional weight molding mechanism aimed at rescaling the connection weights around a set-point [32]. Namely, the classical view of network plasticity ruled on correlation-based mechanisms (such as the above Hebbian rules) is coupled with homeostatic mechanisms that promote network stability through a set-point of network activity that is dynamically maintained. What the main features of this activity are is unclear at present. Rather, biologists highlight different quantities that may be modulated in order to maintain this set-point. For instance, *synaptic scaling*, obtained by scaling the strength of all of a neuron's inputs up or down, determines how fast the neuron fires for given input signals [54]. Differently, *intrinsic plasticity*, obtained by moving the input/output curve of a neuron left or right, allows a neuron to fire less or more for given input stimuli [55]. From a true information perspective we assume this functionality too hard to be committed to a single neuron, hence out of the connectionist paradigm. Rather, we prefer to consider homeostasis as the effect of cyclic evolutions followed by the single neuron endowed with very simple monotonic functionalities and embedded in a milieu capable of volume transmission [56].

With this aim, hiding the synaptic plasticity for the moment, the thresholds of the single neurons play the role of the strength parameter $\gamma$ discussed in Section 2. Again, it represents an age parameter moving from small toward high values. The former denote youngness of the neuron – an age when it is very sensible to external stimuli. This means it fires often under their solicitation, with a side effect of propagating its sensitivity to surrounding neurons (i.e. awakening these neurons). As its age increases the neuron becomes proof against these stimuli, hence less and less prone to fire and to propagate as well, till becoming actually dead w.r.t. the current flow of signals. The aging mechanism determines a long-term homeostasis of the system. We assume that the single neuron increases its threshold by discrete quantities as a reaction to the difference between its age and the mean age of the other live neurons in the layer, i.e. to the difference between its firing rate w.r.t. the mean firing rate.

We may deal with a similar process in an immunitary system. In spite of current evolutionary models where the behavior of bacterial cells is considered at the population level, disregarding the possibility of signals mediating communication between the individual bacterial cells and the rest of the population [57,58], biological evidence for communication phenomena and primitive "decision-making" behavior exists. At the same time, the view of bacterial cells as living beings devoid of individuality is also strongly debated. Asymmetric features are present even in organisms undergoing asexual binary reproduction, such as bacteria [59]: biologists no longer consider heretical the concept of bacterial aging at the individual cell level, as well as phenotype heterogeneity and metabolic asymmetry within a bacterial clone, even in the presence of identical genotypes [60,61]. Individually aging bacteria have been observed in some experiments [59]. This moves (a part of) the population fitness into specific goal functions of the single bacteria.

In particular, consider the evolution of a population of bacteria in the presence of penicillin molecules provided in the culture medium. The distinguishing feature of this scenario is represented by the fact that bacteria may differ in their resistance to antibiotic as a consequence of their differential ability to produce and secrete an enzyme which destroys the antibiotic. The differentiating process between individuals is connected to an irreversible mechanism that we associate with their aging. Like with usual predator-prey models discussed before, the composition of linear behaviors of single individuals may be exactly synthesized through a whole linear function at the population level. Thus, we expect this aging to be connected to a nonlinear behavior and focus on stepwise modifications of a state variable of the individual as a primary source of an attitude that we denote as "intelligence". Discarding approaches that are too much demanding to this intelligence – such as cellular automata [62] or Multi Agent Systems [63] – we look for bacteria computations that are even more elementary than McCulloch and Pitts neuron's [64] in that:

(1) we identify the status of the individual with its output;

(2) we give a trend to the output, thus synthesizing in a single operation the two levels of operations, computing and learning, describing the neuron life, where learning consists of system parameters' modifications;

(3) we allow a one directional trend to the output, thus reducing the intelligence of the cell to the very primordial relation: if solicited, the cell reacts ever in the same way, but this reaction depends on the mentioned thresholding operation.

Coming back to neural networks, the analogous of the reproduction capability of cells is represented by the mentioned awakening ability of surrounding cells. In this case the topology of the awakening connections may induce an interesting phenomena. For instance, the neurons of a same layer may be awakened by the current one, where each layer cooperates in transmitting signals from a previous layer to a subsequent one. Awakening a neuron means reducing its threshold from a high value, which puts it in a quiescent status, to a low value, allowing it to react to the external stimuli. Thanks to this additional functionality, at a higher level the neurons' aging seems to correspond to the overlapping of thought cycles: a set of low threshold neurons primes the cycle by reacting to the external stimuli and pushing surrounding neurons to do the same. We may figure a pre-climax phase where this mechanism is self-excitatory, followed by a post-climax phase where neurons tend to be quiescent and do not awake other ones. Hence the thought cycle ends, with some feeble fallout. The related groups of neurons remain quiescent until they are awakened by neurons from other zones of the brain involving them in another thought cycle.

In this chapter homeostasis combines with the health state of the agents, so that we follow the evolution of the system through sequences of clusters of agents that *are at work*, hence producing items with a good rate, in spite of all environment solicitations discouraging their activity. As a further step in this direction, we also consider agents drawn by many interests (i.e. made up of many components). In this case the healthy homeostatic process has a further degree of freedom, since at each time stroke we check also the relative activity intensity between the various interests, possibly labeling the agent with its prevailing interest. This requires a strength sharing mechanism between the agent components that we will discuss in the last section.

# 4 Modeling and Simulating Our Individuals' Society

To follow the dynamics of the above processes we set up a high-level functional language within the paradigm of $\pi$-calculus [37].

The two objectives we had in mind were:

- to involve very elementary computational tasks to be born in principle by the basic structural and functional units of any organism, each put at most in a bipolar relation with another entity, and
- to implement true parallel algorithms capable of efficiently running on massive parallel computers, where ensemble functionalities are vehiculated by gathering functions that could be implemented by chemico-physical reactions in nature as well.

## 4.1 The $\pi$-Calculus Framework

Both neuronal population and bacterial colony models described in Section 3 fall into the class of concurrent and communicating system. In 1992 Milner [65] published a formal language, the $\pi$-calculus, to describe a general purpose massively concurrent system with intensively communicating facilities, proving that it is a universal language. Its very essential core is described in Table 1.

**Table 1.** Minimal syntax of $\pi$-calculus

| | | |
|---|---|---|
| $P$ :: | $c(m).P$ | Input prefixing |
| $\vert$ | $c\langle n\rangle.P$ | Output prefixing |
| $\vert$ | $P_1 \vert P_2$ | Concurrency |
| $\vert$ | $(\upsilon x)P$ | Restriction |
| $\vert$ | $!P$ | Replication |
| $\vert$ | 0 | Nil process |

Basically, the system is constituted by a series of pairs of reading and writing units located on the extremes of oriented communication channels within a network. A process determines the sequence with which the units execute their operations to pass data along the network. A data item of name $m$ read in input from a channel $c$ may be modified through local computations before being written with the name $n$ in output to the channel. Since the specific timing of each read operation $c(m)$ or write operation $c\langle n\rangle$ depends on the inner dynamics of the hosting computer, any process may be completely specified through the constructs in Table 1, possibly gathering sets of the above operations. Namely, the most elementary ones are the *communication constructs* in the form of *input prefixing* $c(m).P$ and *output prefixing* $c\langle n\rangle.P$. We may read the former as the concatenation of a process $P$ waiting for a message from the communication channel named $c$ before running. The latter synthesizes with $P$ all that is computed by the unit after having both produced $n$ (as concerns internal computational timing) and written it in output to the channel $c$ (as concerns external synchronization timing). In particular, if $n$ is the Nil output (denoted by 0 in Table 1) there is no $P$ continuation since the whole execution is complete and the process is stopped. On the one hand, the iterative implementation of the above constructs gives rise to the general *sequencing* construct $P_1.P_2$ of two processes $P_1$ and $P_2$. On the other, the connection layout may allow the two processes to be executed simultaneously, a condition that is denoted by the *concurrency* (or *parallel*) operator $P_1|P_2$. In particular, the *replicator* operator $!P$ denotes a process $P|P|P|\ldots$ executing simultaneously an infinite set of $P$ copies. Finally, as for unitary constructs, the *restriction* operator $(\upsilon x)P$ bounds the value of $x$ in the process $P$ (it may be seen as a process allocating a new constant $x$ within $P$), while the Nil process induces the execution to stop as explained above. The key operational points of the $\pi$-calculus paradigm are:

(1) the whole process evolves each time a communication on a channel is ready (this means that both a sending and a receiving process are available to respectively writing and reading at the two ends of the channel);

(2) as no distinction is made between variables, data and channels (they are all considered as names), channels themselves can be sent in messages through (other) channels, allowing the topology of process interconnections to change. This renders $\pi$-calculus an elegant language for describing concurrent computations whose configuration may change during the execution of the computation itself.

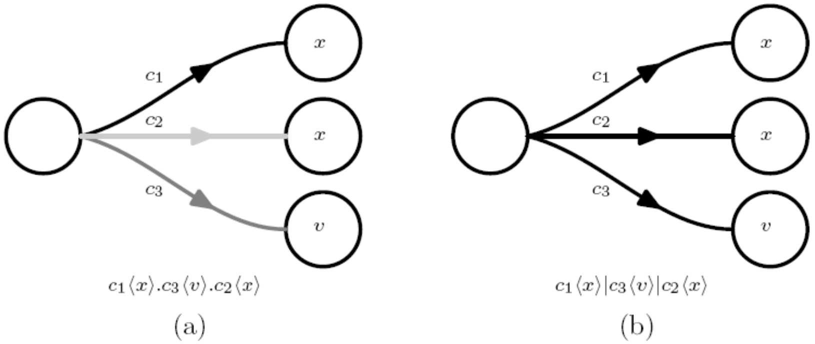

$c_1$
$c_2$
$c_3$
$x$
$x$
$v$
$c_1\langle x\rangle.c_3\langle v\rangle.c_2\langle x\rangle$
(a)
$c_1$
$c_2$
$c_3$
$x$
$x$
$v$
$c_1\langle x\rangle|c_3\langle v\rangle|c_2\langle x\rangle$
(b)

**Fig. 4.** Sequential (a) and parallel multicasting (b) subtended to local communications. (a) Only after sending name *x* on channel $c_1$ to the first receiving process (first unit on the right) and subsequently name $\upsilon$ on channel $c_3$ to the third unit, the sending process (circle to the left in both figures) can transmit name *x* on channel $c_2$ to the second unit; (b) the three units on the right receive simultaneously the names posted by the sending process.

Due to its power and versatility, $\pi$-calculus has become the favorite framework for describing processes in very different fields, from business to biology, and lots of different paradigms and tools based on it have been developed. According to the $\pi$-calculus paradigm, we have an ensemble of binary computations consisting of reactions between two elementary computing elements individually attached at the extremes of a communication channel. Namely on the right of each channel we have an agent (e.g. neuron, bacterium, worker, etc), while at the other extreme we have a transmitting unit (neuron, penicillin, noxious agent, etc.) which may send or not a signal to it. In principle, each element from the list of the former agents is waiting to get in touch with an element of the sending unit list, and *vice versa*. The connection occurs through a set of channels managed by a connection dispenser (in a certain sense, the static version of the Brownian motion). It associates to each neuron not yet filled (in wait status) an element randomly selected from the units on the other list that are available for the specific neuron (hence in wait status as well).

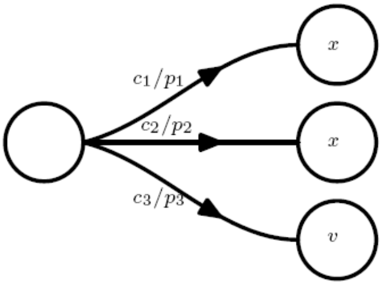
A crucial factor in this dynamic is represented by the reaction time once the connections are stated, i.e. the time to elaborate an item and eventually producing a content, in the workers' acception. On the one hand, the communication is locally synchronous allowing either a sequential or a parallel multicasting (see Figure 4). On the other, we do not want to synchronize the process. Rather, each reaction has its own time. Although we could set reaction times individually, we prefer to connect them directly to the real time employed by the computer, in order to both speed up the entire process and rely on a principled random elapsing time.

A further feature of our process is that each agent may split in two, so giving rise to an awakened offspring, with probability as in (4), which plays the same communication protocol. The branching has in principle an exponential increase that is moderated, when it occurs, by the death of some of the offspring.

To cope with these peculiarities of the agent life we extended the syntax in Table 1 introducing more syntactical objects as summarized in Table 2, representing a variant of stochastic $\pi$-calculus [66].

**Table 2.** Syntax of our stochastic $\pi$-calculus

| | | |
|---|---|---|
| $P ::$ | $c(m).P$ | Input prefixing |
| $\mid$ | $c\langle n\rangle.P$ | Output prefixing |
| $\mid$ | $P_1 \mid P_2$ | Concurrency |
| $\mid$ | $(\upsilon x)P$ | Restriction |
| $\mid$ | $!P$ | Replication |
| $\mid$ | $0$ | Nil process |
| $\Sigma ::$ | $\pi.P + \Sigma$ | Action |
| $\mid$ | $p\rightarrow \pi.P + \Sigma$ | $p$-action |
| $\pi ::$ | $c\langle n\rangle$ | Output |
| $\mid$ | $c(m)$ | Input |



$$\begin{cases} p_1 \rightarrow c_1\langle x\rangle.P_1 \\ p_2 \rightarrow c_2\langle v\rangle.P_2 \\ p_3 \rightarrow c_3\langle x\rangle.P_3 \end{cases} \qquad \begin{cases} c_1(m).P_1' \\ c_2(m).P_2' \\ c_3(m).P_3' \end{cases}$$

**Fig. 5.** According to the probability distribution associated to the channels, the sender selects which one will be used for the transmission of the signal. The receiver waits on all the channels until one becomes available. For instance, if the sender randomly chooses channel $c_2$ then name $\upsilon$ will be passed to the receiver which in turn bounds it to the name $m$. Subsequently the computation proceeds in parallel with processes $P_2$ and $P_2'$.

In particular we explicitly divide the operations carried out by the processes in three categories. We put the most elementary operations $\pi$ represented by the output $c\langle n\rangle$ and the input $c(m)$ in the *actions* category. The other constructs in Table 1 are gathered in the $P$, $Q$ constructs category. At an intermediate level we specify the $\Sigma$ category expressly managing the unpredictability in the choice of multiple actions. Namely, we denote through the *summation* $\pi_1.P_1 + \pi_2.P_2 + \ldots + \pi_n.P_n$ the condition where more channels are allowed through which to pipe signals and one of such $\pi_i$ actions followed by the related $P_i$ construct will surely occur, but we do not know which one. The actual continuation of the execution depends on the channel availability and possibly on a specific distribution probability. In other words, the construct $\pi.P + \Sigma$ executes the action $\pi$ at the end of the channel available at first, followed by its $P$ suffix. *Vice versa*, the construct $p\rightarrow \pi.P + \Sigma$ preliminarily tosses the channel according to a probability distribution (see Figure 5), then waits until this channel is available and executes the related $\pi$ action when it happens.

With this formalism, the two biological systems considered in Section 3 may be quickly modeled as follows.

### 4.2 The $\pi$-Calculus Bacterial Model

We implement the evolution of the bacteria-penicillin molecules population through the update of 3 quantities: $n$, $m$ and $\gamma_B$, as follows:

1. set of living bacteria. Starting from an initial set of $n(0)$ bacteria, the set either increases or reduces depending on the fate of the single individual;
2. set of penicillin molecules (p-molecules) within a liquid medium, accompanied by a set of non-penicillin molecules (np-molecules). Starting from a set of $m(0)$ p-molecules within a set of $L$ molecules, hence a set of $\overline{m}(0) = L - m(0)$ np-molecules, a p-molecule may shift into an np-molecule if successfully attacked by the penicillinase; the inverse shift will never happen.
3. strength $\gamma_B$ of the single bacterium. B decides incrementing $\gamma_B$ of a quantity $\delta$ if the actual value of $\gamma_B$ is greater than the average $\bar{\gamma}_B$ over all the colony by a factor less than a threshold $\tau$, according to (3).

The evolution of these quantities is linked to the fate of the single agents that, according to (1-4), specializes as a consequence of two independent random events, related to the inner state of the bacterium and to the environment conditions, respectively. Namely, B either splits ($S$(B) = 1) or emits penicillinase. At the same time B either meets a p-molecule ($M$(B) = 1) or an np-molecule. In addition, if $M$(B) = 1 then the bacterium may die ($D$(B) = 1) with a probability that depends on whether it is splitting in two (hence is weaker against the attack) or not. As a conclusion B either dies or splits, or remains as it is. Namely,

$$P(S(B)=1) = 1 - \gamma_B \tag{8}$$

$$P(M(B)=1) = \rho \tag{9}$$

$$P(D(B)=1) = \begin{cases} \rho\mu_S \ \ if \ S(B)=1 \\ \rho\mu_E \ \ if \ S(B)=0 \end{cases} \tag{10}$$

where $\rho = m(t)/L$ is the penicillin density in the solution, and $\mu_S$ and $\mu_E$ are mortality coefficients.

In own turn, the penicillin density $\rho$ decreases of a quantity $1/L$ with a probability $\mu_R$ each time an individual that is emitting penicillinase and is attached by penicillin does not die. Namely: at each time step a bacterium is invested by a penicillin molecule with probability $\rho$. Then we have a two-level tree: at the first level a bacterium either emits penicillinase (with probability $\gamma_B$) or duplicates (with probability $1 - \gamma_B$). In the first option it either dies with probability $\mu_E$ or destroys a penicillin molecule with probability $(1 - \mu_E)\,\mu_R$.

$$\nu\gamma\ !\mathrm{Bat}(\gamma)\begin{cases}\gamma & \begin{cases}\mathrm{Attack}(); & \begin{cases}\mu_E & \mathrm{PenLive}\langle\rangle\\ 1-\mu_E & \begin{cases}\mu_R & \mathrm{PenDie}\langle\rangle;\mathrm{Bat}\langle\gamma\rangle\\ 1-\mu_R & \mathrm{PenLive}\langle\rangle;\mathrm{Bat}\langle\gamma\rangle\end{cases}\end{cases}\\ \mathrm{NoAttack}(); & \mathrm{Bat}\langle\gamma\rangle\end{cases}\\ 1-\gamma & \mathrm{Bat}\langle\gamma\rangle; \begin{cases}\mathrm{Attack}(); \begin{cases}\mu_S & \mathrm{PenLive}\langle\rangle\\ 1-\mu_S & \mathrm{PenLive}\langle\rangle;\mathrm{Bat}\langle\gamma\rangle\end{cases}\\ \mathrm{NoAttack}();\mathrm{Bat}\langle\gamma\rangle\end{cases}\end{cases}$$

$$!\mathrm{Pen}();\mathrm{Attack}\langle\rangle;\begin{cases}\mathrm{PenLive}();\mathrm{Pen}\langle\rangle\\ \mathrm{PenDie}();\mathrm{NonPen}\langle\rangle\end{cases}$$

$$!\mathrm{NonPen}();\mathrm{NoAttack}\langle\rangle;\mathrm{NonPen}\langle\rangle$$

**Fig. 6.** Stochastic $\pi$-calculus program describing the interaction between bacteria and penicillin molecules

The language components to implement this dynamics on a $\pi$-calculus are: a *Bat* channel describes one bacterium in the medium, and the corresponding rate identifies with the bacterium's strength. Idem for the *Pen* channel, related to penicillin molecules. A penicillin attack to a bacterium is featured through the *Attack* and *NoAttack* channels. A successful attack gives rise to a *PenLive* channel which in turn maintains unchanged the number of penicillin molecules in the medium. Analogously, an unsuccessful attack is modelled through a *PenDie* channel and a subsequent replacement of the original *Pen* channel with a *NonPen* one. Figure 6 describes the prototypical behaviour of three different components of the system: bacteria, penicillin and non-penicillin molecules using the syntax of the stochastic $\pi$-calculus introduced in Table 2 with two further notations: i) we substitute the plus in the summation operator with curly brackets, for the sake of simplicity, and ii) the input/output actions may be preceded by names denoting their probabilities.

Initially a bacterium, with a probability equal to its strength $\gamma_B = \gamma$ emits penicillinase, otherwise duplicates. In both situations it waits uniformly on a channel *Attack*() or *NoAttack*() until a p-molecule or an np-molecule decides either to attack or not the bacterium. Differently from penicillin, which always attacks a bacterium, non-penicillin plays a passive role to guarantee a continuous evolution of the system also when a low density of p-molecules remains in the culture. In case no attack is delivered, the two communicating processes end the current iteration and recall an individual of the same kind (for instance a bacterium activates the channel $Bat\langle\gamma\rangle$) which will begin a new life-cycle. Otherwise, if penicillin attacks the bacterium, the fate of the latter will depend on a mortality coefficient whose value changes according to the action previously performed by the bacterium. So, if the latter had reproduced, it dies with a probability $\mu_S$ (and consequently it does not recall itself), while in both cases it activates the channel $PenLive\langle\rangle$ to inform the attacking penicillin to activate a new copy of itself. If the bacterium had emitted penicillinase, then its death probability lower to a value $\mu_E$: what happens is similar to the previous situation, with the

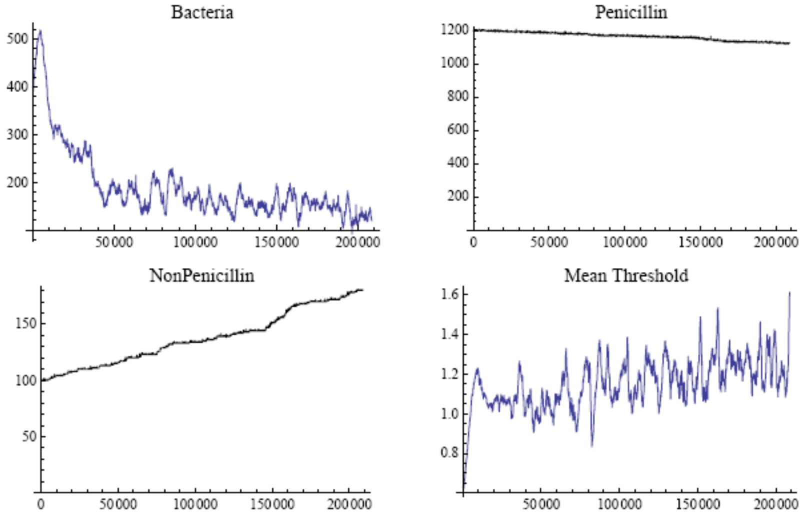


**Fig. 7.** Course of main quantities of a *quasi* stable match between bacteria and penicillin molecules with the biological clock

exception that the p-molecule may be destroyed (with a probability equal to $\mu_R$, the mortality coefficient for penicillin molecules) if the bacterium survives to the penicillin attack. If this happens, the penicillin will be informed about its own death through a message on the *PenDie*() channel. We need some further technicalities to make the process correctly evolving that the interested reader may find in the original paper [67]. Here we report show three pictures describing:

a. an homeostatic condition (see Figure 7) denoted by a substantially stable number of bacteria as a fallout of a gentle decreasing of the penicillin molecule density and a gentle increasing of the mean strength. Actually, we cannot exclude the implosion of bacterial population after a certain time. However given the employed parameters, we may assume the simulated period to correspond to 2–3 days of an actual biological system.
b. a non stable condition leading to the explosion of the bacterial population (see Figure 8) due to a gently increase of mean strength that does not prevents a tangible bacterial reproduction at each time stroke. This denotes a non suitable balancing between the penicillin efficacy and the value of the single strength increment. Note that a sharp increase of strengths causes, on the contrary, the bacterial implosion. In this case indeed, we obtain only bacteria capable of defending hard themselves, but, since they are not able to replace deaths, they are destined to the extinction.
c. the joint evolution of two bacterial populations, i.e. different genotypical versions of the same bacterium. Base histories are the same for each population, but they interfere in a complex way. Essentially, we have an initial overwhelming of the strongest population but a long term revenge of the weakest one. In these experiments we distinguish the two populations by an

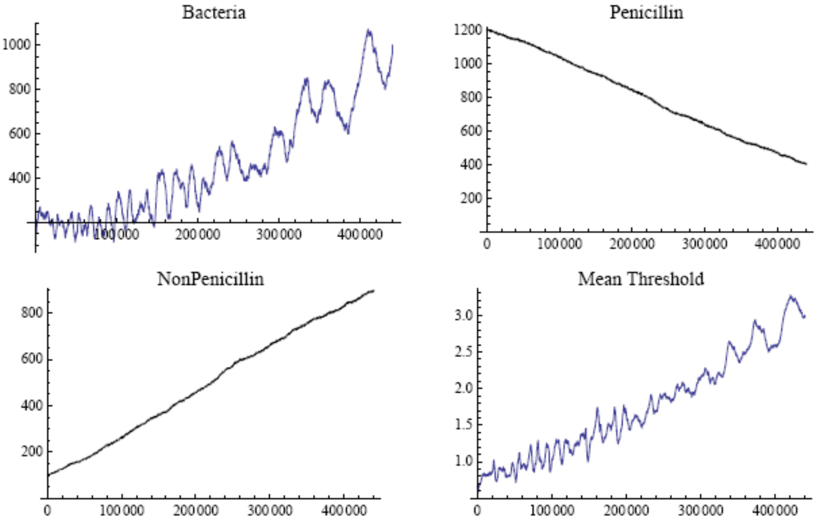


**Fig. 8.** A population going to explode. Same notation as in Figure 7.

initial strength distribution, biased toward lower values with the latter, and a less aggressive strength updating policy (i.e. smaller $\delta$) again with the latter. All other parameters are the same. We have a first phase where the stronger population dominates the latter getting higher values to their strength and leaving the reproduction act mainly on the weakest population. After a first almost stable plateau (see first picture in Figure 9), the first population individuals die, although they had reached greatest strengths (red line in the last picture), thus giving completely camp to the other population. At genotypical level we may imagine that strongest individuals are renewed by the long term mutation process along the generations, giving rise to a sequence of variants' alternations like the ones described in Figure 9. We cannot refrain from noting how a similar behaviour could fit other operational fields, as usual for $\pi$-calculus applications. For instance, in macroeconomy, the two populations could represent different socio-economical classes both fighting against an hostile environment [68].

A good perspective on these processes comes from the strength histograms. Figure 10(a) reports the two typical features of strength distributions for the single population depicted in Figure 7. Namely: i) a quantization of the strengths, so that only some ranges of values are allowed, and ii) the Gaussian spread of the quantized values around different centers, with the global effect of observing a mixture of normal distributions. In own turn this mixture gives rise to a bimodal overall distribution, with one narrow peak close to the maximum and a shallow one centered slightly under the central value of the current strength range. In case of metastable equilibria, the fate of the bacterial population depends on the position of the latter center, that causes explosion when biased toward zero, implosion otherwise. Coming to the double population scenario of Figure 9, in spite of the initial

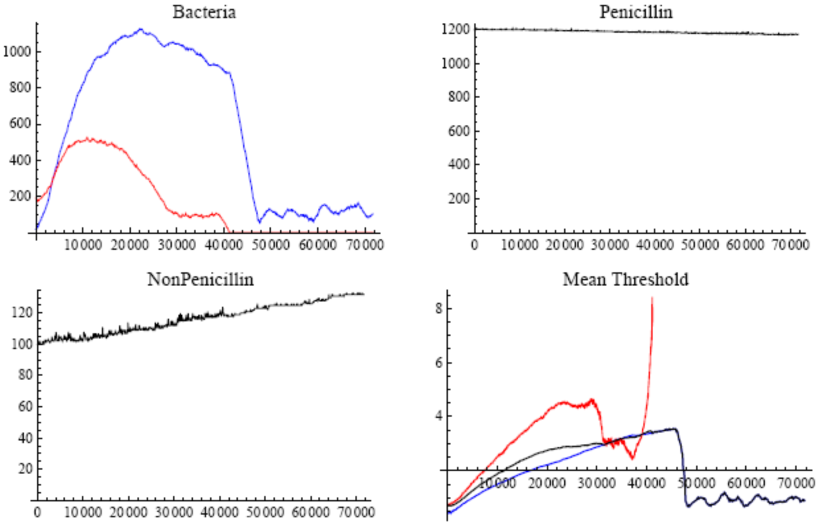
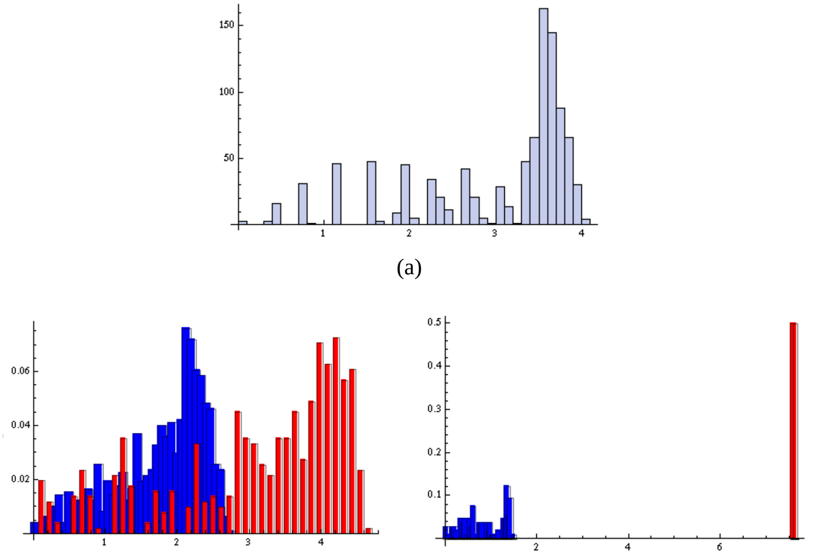


**Fig. 9.** A merge of two bacterial variants. Red lines refer to the strongest population, blue lines to the weakest one. Black line in the strength diagram refers to the mean value between the populations.



(a)

(b)

(c)

**Fig. 10.** Histograms of the bacterial strengths in metastable conditions: (a) for a single population; (b) at the starting of the joint evolution of two populations; and (c) at the stable prevailing of one of them

distributions (see Figure 10(b)), in Figure 10(c) we recover the same features on the weakest population, while, just before dying, the second population is concentrated around very high values that prove unable, however, to avoid extinction.

### 4.3 The *π*-Calculus Neuronal Model

An analogous calculus may be implemented for instance on a multilayer neural network where the signal flows from the uppermost to the lowermost layer. In relation to their mutual positions we identify any pair of consecutive layers as the upward layer feeding the downward one through a set of connections. In this way each layer is both an upward one with respect to a pair and a downward one w.r.t. another pair, with the obvious exception of the extreme layers. With the local awakening facility introduced in Section 3, we may be interested in following the two patterns: i) the number $m$ of spikes produced by the downward layer neurons (for short d-neuron) in a temporal window, and ii) the local distribution of neurons producing the spikes. In analogy with bacterial population we describe the evolution through the following points:

(1) set of d-neurons. They refer to any downward layer. In addition to the connections with the upward layer, these neurons are connected to the others in the same layer through an incomplete matrix. As a first approximation we fix connections randomly, by assuming each neuron connected with the others of the layer with a given probability $\rho$, the latter being a function of the distance of the neurons on the grid. We distinguish live neurons from dead ones, where the latter may be either physiologically or functionally dead. In any case, a neuron never producing a spike in a given temporal window is considered dead. A functionally dead neuron becomes live if primed by a spiking neuron along an existing connection;

(2) set of u-neurons. They refer expressly to any uppermost layer. Each neuron possibly sends a unitary signal to the downward neurons depending on the evolution of the whole brain state, provided that a connection exists between the unit and the receiver neuron. As a preliminary approximation, the topological map describing the inter-connections between couples of neurons in two different layers is fixed a priori and depends on the distance of the paired neurons. A signal is sent from an upward neuron to a receiver neuron with a probability $b$. In case the former belongs to the top layer of the network, $b$ is a constant to keep the model as simple as possible. Otherwise $b$ is related to the firing probability of the connected upper neuron. Analogously, the connection weight has a random value $w$ in a given interval centered around 0;

(3) fate of the single d-neuron. It depends on its basic functionality of computing the heaviside function of its potential $v$. Namely, at each instant a d-neuron B tries to communicate with an upward neuron, say $j$, receiving a non null signal $X_j(\mathrm{B})=1$ with the above probability $b$. The transmitted signal $w_j X_j$ is null with probability $1 - b$, otherwise coincides with the weight $w_j$ of the link. If $v + w$ trespasses the threshold $\gamma_B$, then the neuron fires ($S(\mathrm{B})=1$) and sets its potential to the resting value $v_0$. Otherwise

$v=v+w$, or a smoothed form of this sum to take into account the decay of the signal over time, for instance:

$$v = v + w - \beta(v - v_0) \tag{11}$$

with $\beta$ playing the role of decay rate. If the neuron fires, then there is a probability $\rho\alpha^t$ that it awakes another d-neuron B′ connected to it according to the connection matrix as in Point 1, getting $L(\mathrm{B}')=1$, where $L$ plays here the role of living indicator. The first term $\rho$ takes into account the connectivity of the neurons, while $\alpha < 1$ is a dumping term having the effect of reducing the awaking probability of a neuron B′ that has already been risen $t$ times. Alternatively, $L(\mathrm{B}')=0$ in the case of either physiological or functional death. Namely, hiding the indices the boiling up probabilities are

$$\mathrm{P}(X(\mathrm{B})=1)=b \tag{12}$$

$$\mathrm{P}(S(\mathrm{B})=1)=F_W(v-\gamma_B) \tag{13}$$

$$\mathrm{P}(L(\mathrm{B})=0)= \rho\alpha^t \tag{14}$$

where $F_W$ is a heavyside function;

(4) strength $\gamma_B$ of the single neuron. According to (3), whenever B fires, it decides to increment $\gamma_B$ by a quantity $\delta$ provided that the actual value of $\gamma_B$ is no greater by a fixed $\tau$ than the average $\bar{\gamma}_B$ over all the live d-neurons it is connected to. The name of the neuron is B to evoke Bob in the previous game; funny to say, each production of B is to be considered like a Bob defeat requiring the strength increase;

(5) the localization of the live neurons. It is determined by the connection matrix. Within a given neighborhood we may still maintain the connection uniformly random from upward to downward layers. An analogous randomness on the intra-layer connection between d-neurons will produce radial attraction basins around the neurons that initially start a thought. Instead, more complex figures may emerge from different settings of the connections. The connection matrix, along with the position of the priming neurons, will generate a spatial process of thought with possible overlapping and alternating phenomena inside the layer.

The functionalities considered may be specified through the three *metaconstructs*: **Uneu**, **Dneu** and **Dead** described in Figure 11. There, we give the layout of the constructs according to the additional notation: 1) computations from input to output of a channel are described inside the angle brackets of the output action; 2) since no iterative construct is permitted in $\pi$-calculus, recursiveness is implemented on a channel by duplicating it, i.e. by putting it as both terminal of a metaconstruct and first channel of another – possibly the same – metaconstruct which starts with the parameters piped by this channel (see discussion on how to code a new bacterial life-cycle in $\pi$-calculus after Figure 6).

- **Uneu.** This construct figures the uppermost neurons. Each neuron is associated through the restriction operator $\upsilon$ with both a name *b* denoting the probability of sending a signal according to (12), and an implicit name *w*, whose value depends on the underlying network topology, giving the strength of the transmitted signal. As replication operator $!\pi.P$ defines multiple copies of $\pi.P$, a certain (possibly infinite) number of processes are waiting on the *Unit* channels for receiving a triggering signal carrying the name *b*. Only when such a communication takes place, the unit sends a signal *w* through a synapse (described by the *Syn* channel) with probability *b*. Otherwise, i.e. with probability $1-b$, it posts a fictitious signal on a dummy synapse represented by the *NoSyn* channel. *Unit* channels end the construct as well allowing the process to cycle, as explained before.

$$\mathbf{Uneu} :: \nu(b,w)\ !\mathit{Unit}(b).\begin{cases} b \rightarrow \mathit{Syn}\langle w\rangle.\mathit{Unit}\langle b\rangle \\ 1-b \rightarrow \mathit{NoSyn}\langle\rangle.\mathit{Unit}\langle b\rangle \end{cases}$$

$$\mathbf{Dneu} :: \nu(\gamma,v,s,h,w)\ !\mathit{Neur}(\gamma,v,s,h).\begin{cases} al(h) \rightarrow \begin{cases} \begin{cases} \mathit{Syn}(w).\begin{cases} F_W(v+w-\gamma) \rightarrow \ \dots \\ 1-F_W(v+w-\gamma) \rightarrow \ \dots \end{cases} \\ \mathit{NoSyn}().\mathit{Neur}\langle \gamma, v(1-\beta)+\beta v_0, s, h-1\rangle \end{cases} \\ \mathit{Reset}(\gamma).\mathit{Neur}\langle \gamma, v, s, h_{\max}\rangle \end{cases} \\ 1-al(h) \rightarrow \mathit{Awake}(\gamma,\_,\_,h_{\max}).\mathit{Neur}\langle \gamma, v, s, h-1\rangle \end{cases}$$

$$\dots\begin{cases} \rho\alpha^t \rightarrow \begin{cases} \eta & \mathit{Awake}\langle \mathit{rand}[\gamma_1,\gamma_2], v_0, 0, h_{\max}\rangle \\ 1-\eta & \mathit{Reset}\langle \mathit{rand}[\gamma_1,\gamma_2]\rangle \end{cases} |\mathit{Neur}\langle \gamma+\delta\gamma, v_0, s+1, h_{\max}\rangle \\ 1-\rho\alpha^t \rightarrow \mathit{Neur}\langle \gamma+\delta\gamma, v_0, s+1, h_{\max}\rangle \end{cases}$$

$$\dots\mathit{Neur}\langle \gamma, v(1-\beta)+\beta v_0+w, s, h-1\rangle$$

$$\mathbf{Dead} :: \nu(\gamma,v,s,h)\ !\mathit{Awake}(\gamma,v,s,h).\mathit{Neur}\langle \gamma, v, s, h\rangle$$

**Fig. 11.** Stochastic $\pi$-calculus model for neuronal interactions

- **Dneu.** This construct describes the behavior of the d-neurons. In order to implement (13) and (14), we need to fix through the restriction operator: 1) *w* from the connection matrix, 2) the initial values of the threshold $\gamma$ and potential *v* randomly, 3) the number of emitted spikes *s* to 0, and 4) the latency parameter *h* to the value $h_{\max}$. Then we cycle the current values of $\gamma$, *v*, *s* and *h* through the channels *Neur* at the beginning of the metaconstruct and the ends of some branches. According to the probabilities (12-14), the process evolves along five branchings depending on: 1) the neuron is alive or not; in the first case, 2) it may receive a signal from neurons in the same layer (*Reset*) or from the upward layer; 3) in the second case, it may receive or not receive a signal; in the first case, 4) it may fire or not, and 5) if it fires it may awake dead neurons in the province of the first branching. In order to complete the dynamics the neurons are endowed with constants ($\rho,\alpha,\beta,\eta$), respectively representing the coefficient and dumping term in (14), a smoothing factor in (11), and the awakening probability. According to $\eta$ the neuron awakening occurs in a twofold form: with a probability $\eta$ a functionally dead neuron (waiting on an *Awake* channel) is awakened by sending a signal on the same channel, while with probability $1–\eta$ the threshold value of an active unit is lowered by sending a signal on a *Reset* channel. The latency parameter $h$ is updated so as to account on each neuron for the number of potential updates before firing. Namely, it is set to an $h_{\max}$ value at each firing and diminished by 1 at each non firing update. A neuron is considered functionally dead if its latency parameter is equal to 0; in this case $al(h)$ equals 0 otherwise 1.

- **Dead.** This is the construct of inactive d-neurons. They wait for an *Awake* channel to receive a suitable threshold allowing it to be reactive to the u-neuron signals. Both *Awake* and *Reset* channels fix the new thresholds to random values in a given interval.

Figure 12 reports the four-layer architecture aimed at simulating a portion of the thalamocortical circuit of a rat brain (in particular relay cells thalamus layer, interneurons thalamus layer, neocortical layer V and neocortical layer IV respectively) [69]. The final target of the research project is to reproduce some local coherence phenomena occurring in these slices as a consequence of a state of chronic pain in a rat. Neural signals are registered in an experimental setup through a set of probes inserted in the rat scalp [70]. At the moment we just exhibit in Figure 13 some temporal patterns emerging in the artificial layers as a consequence of the neighboring relations stated between neurons as mentioned earlier. The rows are joint frames taken from a simulation strip at appropriate times. Namely the former one refers to an initial time when neurons are highly decorrelated; the latter highlights a tangible structure conditioning the firing of group of neurons. Thus the health notion here is comprehensive of both the number of firing neurons and their structured aggregation. As a technical detail,

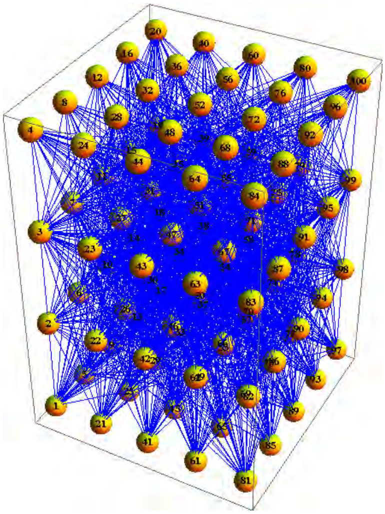


**Fig. 12.** A four-layer architecture simulating a portion of thalamocortical circuit. Each layer is made up of 125 neurons.

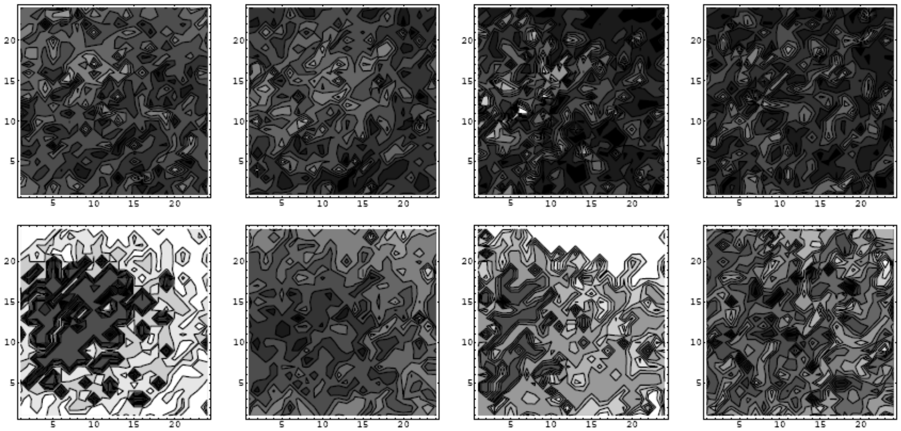


**Fig. 13.** Joint frames from thalamocortical simulation. First row refers to an early stage; second row to a late evolution where some structural patterns become evident. Gray levels refer to spike frequency of the underlying neuron. Contour lines identify the spiking frequency levels.

we mention that we used the Izhikevich model [71] for ruling the neuron firing. For a more deep discussion on the matter we refer the reader to [72].

## 5 A Dynamic Fuzzy Clustering

Consider a population of agents that are mobile like bacteria and are able to awake surrounding people like neurons. For instance we may focus on an opportunistic network [73] supported by a group of communities insisting on a population of individuals, parts of which share the interests characterizing the single communities. The *opportunistic network* paradigm benefits from an extended mouth suggestion facility. Each person may communicate with any other within a certain range – say the one determined by an ancillary radio device – either in flooding mode (pure broadcasting) [74] or in a selective mode (only to those sharing a community interest) [75]. In turn each individual may allow reception of other messages in the given range either indistinctly or selectively as before. We may assume $\gamma$ to be a disaffection parameter pushing the agent not to participate in a chat, as he already knows almost everything about the community and the topic. With this special kind of work (chatting – not so infrequent today) enrolling is directly connected to production: forwarding a message you may awake a person who decides to join the community. In this case we have a dynamic formation of a cluster of live networks denoting a community. Actually we consider a simultaneous activation of more communities. This occurs via a replication of agents, one for each interest, with special connections between the replicas in order to normalize their $\gamma$s, see Figure 14.

Namely, denoting with $A_i$ an integrated agent as the union of more specific ones $\{A_{i1},\ldots,A_{ik}\}$ – let us call them *components*, in addition to (12-14) that we use in an indistinguishable way for both neurons and agents, we have a normalization operation

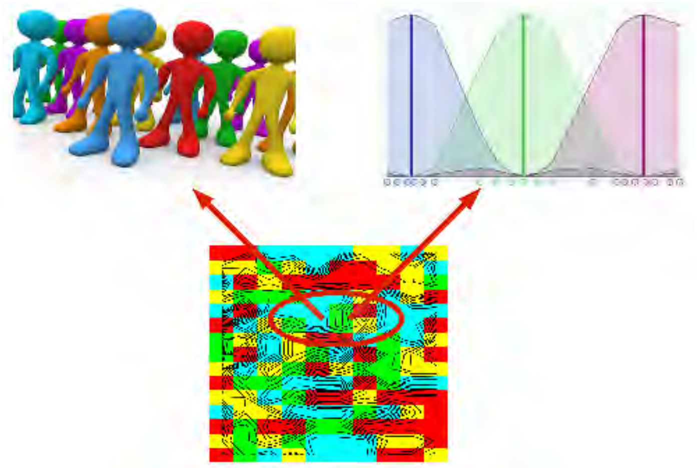


**Fig. 14.** Blow up of a virtual multicommunity. Colors label the communities. Each agent is composed of four components, each with a different color and a different membership to the related community.

$$\mathbf{SNeu} :: \ \nu(b,w)!S(b).\begin{cases} b \rightarrow \ \mathit{Syn}\langle w\rangle.S\langle b\rangle \\ 1-b \rightarrow \ \mathit{NoSyn}\langle\rangle.S\langle b\rangle \end{cases}$$

$$\mathbf{DNeu} :: \nu(v,w,\gamma,\overline{\gamma})!A(v,\gamma,\overline{\gamma}).\begin{cases} \mathit{Syn}(w) \rightarrow \ \begin{cases} \mathrm{HS}(f(v,w)-30) \rightarrow \ldots \\ 1-\mathrm{HS}(f(v,w)-30) \rightarrow A\langle f(v,w),\gamma,\overline{\gamma}\rangle|\mathit{NoSyn}\langle\rangle \end{cases} \\ \mathit{NoSyn}() \rightarrow \ A(f(v,0),\gamma,\overline{\gamma})|\mathit{NoSyn}\langle\rangle \\ \mathit{Update}(\gamma^*) \rightarrow \ g(\gamma^*,\overline{\gamma}) \end{cases}$$

$$\ldots\begin{cases} \mathrm{HS}(\gamma-c\overline{\gamma}) \rightarrow \quad A\langle v_0,\gamma,\overline{\gamma}\rangle|\mathit{Syn}\langle w\rangle \\ 1-\mathrm{HS}(\gamma-c\overline{\gamma}) \rightarrow \quad A\langle v_0,\gamma+\delta\gamma,\overline{\gamma}\rangle|\mathit{Syn}\langle w\rangle|\mathit{Update}\langle\gamma+\delta\gamma\rangle \end{cases}$$

**Fig. 15.** Stochastic $\pi$-calculus model for multi-component agents interactions

that is activated at each $\gamma$ change. Namely, assuming that the $j$-th component $A_{ij}$ increased his strength, we update all components as follows:

$$\gamma_{ij} = \gamma_{ij} + \delta\gamma \tag{15}$$

$$\gamma_{ir} = \gamma_{ir} + \frac{\delta\gamma}{k-1}, \quad r = 1,\ldots,k \ \ r \neq k \tag{16}$$

Thus the p-calculus model for this instance is a bit more complex than for neuron populations, as shown in Figure 15. Here you can see a **SNeu** unit, whose purpose is only to provide a minimal background activity to the main processes, together with a **DNeu** unit, representing the behavior of the single component $A_{ij}$. The general scheme is similar to the one described in Figure 11 although with some noticeable differences. First of all the neuron life-cycle is completely synthesized by the function $f(v,w)$, with $v$ the neuron potential and $w$ the signal received on a synapse. Neuronal models such as Izhikevich [71] – the one currently used in our simulations – Integrate and Fire [76], or even more biologically plausible models [77], can be used.

After a neuron firing, i) a new signal is propagated to the neighboring units through the channel *Syn*, ii) the threshold $\gamma$ is increased by a quantity $\delta\gamma$ whenever it is not greater than the average strength $\bar{\gamma}$ multiplied by a constant $c$, and iii) in the latter case the new threshold value is propagated to the neighboring neurons through the channel *Update*, which in turn will be used to update the value of the overall mean strength $\bar{\gamma}$ through a suitable function $g$ computing, for instance, a mobile average. Continuous time evolution is ensured, as usual, through fictitious message sending over the *NoSyn* channel.

This model is at a very early stage. Here we just forced two typical phenomena we may expect in a group of virtual communities: i) the pervasivity of a trivial thought, say a political slogan or a TV series, and ii) the isolation of an out-of-box thought.

Row (I) in Figure 16 shows three frames of a log concerning the process underlying the former phenomenon. We start with a well variegated situation where integrated agents are homogeneously involved in four interest groups (associated with the colors red, green, yellow and blue in the picture). We may perceive it from the color distribution, since each pixel, i.e. each integrated agent $A_i$ is labeled by the group within which he produces a greater number of outputs (the number of firings in the neuron terminology) per time unit. *Vice versa*, at the end of the process almost all agents are producing with greater frequency within a same interest group. We induced this phenomenon just by pushing each agent to awake a surrounding component involved in this interest group each time he awakens a component of another interest group.

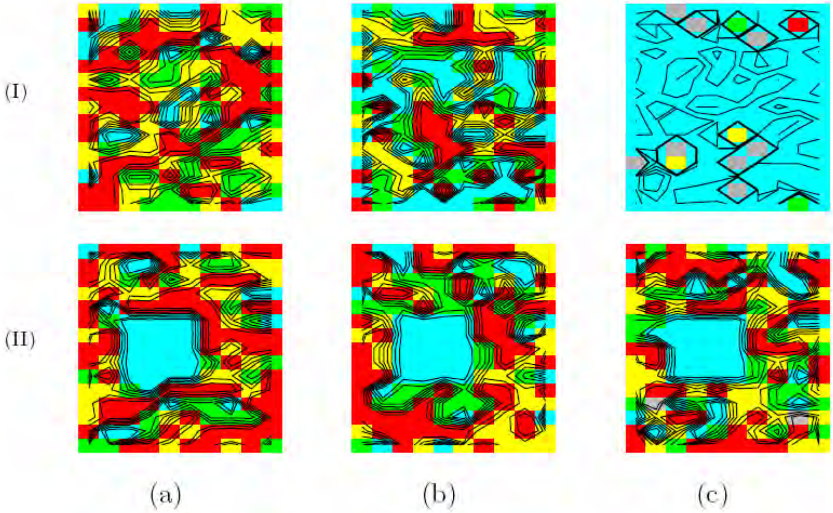


**Fig. 16.** Community clusters within a population. A pixel color highlights the community component whose spiking rate is prevailing. Contour lines denote the frequency levels. Columns denote different evolution times. Row (I) shows the temporal prevailing of a community. Row (II) shows the self-reference behavior of a community.

Row (II) in Figure 16 uses the same tools to describe the second phenomenon. In this case we note a sort of trench around a specific interest group that remains isolated and strictly localized. We induce this second effect by inhibiting the awakening of components of this group. Each time one component of this group should be awakened, another component of the same integrate agent is awakened in his place.

## 6 Conclusions

A keen feature of human-centric information processing emerging from the above discussion concerns whether it is pro or con the individual person. In recent years, great attention has been devoted to stating a synergy between the communication features of Calculus of Communicating System (CCS) languages, such as $\pi$-calculus, and the semantic power of the agent paradigm. The goal is to have an ensemble of agents who interact with one another through communication channels and who are singularly capable of changing their profile in reaction to these communications. Examples of this paradigm may be found in business process design and verification [78,79], web service delivery [80,81], biological cooperation [82], and so on. However, it may seem to be a too ambitious goal, probably destined to the same kind of failure, in the general instances, as parallel computation in the 1980s. Even though the overall organization of the computation is less structured in the present target, we cannot expect a complex coordination between the complex computational tasks of the agents without an even more complex communication protocol, where these two complexities have a reciprocal excitatory effect.

On the contrary, we opted for a low profiling of the agents, considering them just like PEs. This seems to be the same strategy that many market companies and political administrations pursue*: give agents a few facilities and the feeling of caring about their interests through an immediate usage of the facilities. Meanwhile take any opportunity from the combined output of properly biased agent activities.* A common feature of the processes we have dealt with in this chapter is, however, the hardness of their control. Each time we try to have homeostasis represents a new time. We spend hours finding the right parameters, whereas we suspect Nature took millions of years. In a way this is the revenge of complexity and social security: we simplified to the maximum extent the computing duties of the single agents, so we have some difficulty considering them intelligent. Then on the one hand we are ensured that such an army of stupid soldiers may win any kind of war. On the other, finetuning their rules proves to be a very complex task requiring long simulation sessions. This leaves room for another structural optimization problem in search of the breakeven point between complexity of the agent tasks and complexity of the rules assembling them.

All this serves the purpose of the “con individual person” option. On the other hand, the fact that a control law is difficult to fix from abroad does not prevent the PE ensemble from self-organizing itself [83]. This looks for an emergence of true human-centric functionalities that start to appear in various communities focused on local interests, political elections included. In these instances the breakeven problem between complexity of the single agents and their organization may play the role of the leverage $\gamma$ we considered in this chapter.

## References

1. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. John Wiley & Sons, Hoboken (2007)
2. Olsen, D.R.: Interacting in chaos. Interactions 6(5), 42–54 (1999)
3. Roy, A.: Oh sure, my method is connectionist too. who said it's not. In: FSDK 2002, Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery: Computational Intelligence for the E-Age, Orchid Country Club, Singapore, November 18-22, 2002, vol. 2, p. 12 (2002)
4. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. Wiley-Interscience Series In Discrete Mathematics And Optimization. Wiley, Chichester (1999)
5. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability. Springer, Heidelberg (1999)
6. Ayachitula, N., Buco, M., Diao, Y., Maheswaran, S., Pavuluri, R., Shwartz, L., Ward, C.: It service management automation - a hybrid methodology to integrate and orchestrate collaborative human centric and automation centric workflows. In: IEEE International Conference on Services Computing, 2007. SCC 2007, pp. 574–581 (2007)
7. Redbooks, I.B.M.: Human-centric business process management with websphere process server v6. IBM Corp., Riverton, NJ, USA (2007)
8. Przelaskowski, A.: Computer-aided diagnosis: From image understanding to integrated assistance. In: Pietka, E. (ed.) Information Technologies in Biomedicine, vol. 47, pp. 44–56. Springer, Heidelberg (2008)
9. Doi, K.: Computer-aided diagnosis in medical imaging: historical review, current status and future potential. Comput. Med. Imaging Graph 31(4-5), 198–211 (2007)
10. Creese, S., Goldsmith, M., Roscoe, B., Zakiuddin, I.: Research directions for trust and security in human-centric computing. In: Privacy, Security and Trust within the Context of Pervasive Computing, Netherlands. The Springer International Series in Engineering and Computer Science, vol. 780, pp. 83–91. Springer, Heidelberg (2006)
11. Mostéfaoui, G.K., Brézillon, P.: Human-centric network security management: a comprehensive helper. In: WMASH 2006: Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots, pp. 47–52. ACM, New York (2006)
12. Badre, A., Levialdi, S., Foley, J., Thomas, J., Strohecker, C., De Angeli, A., Ram, P., Ram, A., Sanchez, J.: Human centric E-learning and the challenge of cultural localization. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4663, pp. 690–691. Springer, Heidelberg (2007)
13. Pautasso, C., Alonso, G.: Visual composition of web services. In: HCC 2003: Proceedings of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments, Washington, DC, USA, 2003, pp. 92–99. IEEE Computer Society Press, Los Alamitos (2003)
14. Bargiela, A., Pedrycz, W.: Granular Computing, An Introduction. Kluwer Academic Publishers, Boston (2003)
15. Costabile, M.F., Fogli, D., Fresta, G., Mussio, P., Piccinno, A.: Virtual workshops for human-centric computing. In: VLHCC 2004: Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing, Washington, DC, USA, 2004, pp. 65–68. IEEE Computer Society, Los Alamitos (2004)

16. Terziyan, V.Y., Kononenko, O.: Semantic web enabled web services: State-of-art and industrial challenges. In: Jeckle, M. (LJ) Zhang, L.-J. (eds.) ICWS-Europe 2003. LNCS, vol. 2853, pp. 183–197. Springer, Heidelberg (2003)
17. Jackson, P.: Introduction to Expert Systems, 1st edn. Addison-Wesley, Reading (1986)
18. Schuler, D.: Social computing. Communication of ACM 37(1), 28–29 (1994)
19. Johnson, S.: Emergence: the connected lives of ants, brains, cities and software. Scribner (2002)
20. Gustafson, J.L.: Reevaluating amdahl's law. Communications of the ACM 31, 532–533 (1988)
21. Frederick, P.B.: The Mythical Man-Month: Essays on Software Engineering, 20th anniversary edn. Addison-Wesley Professional, Reading (1995)
22. Rumelhart, D.E. (ed.): Parallel Distributed Processing, vol. 1. MIT Press, Cambridge (1986)
23. Small, G., Vorgan, G.: iBrain: Surviving the Technological Alteration of the Modern Mind. Collins Living (2008)
24. Apolloni, B., Bassis, S., Malchiodi, D., Witold, P.: The Puzzle of Granular Computing. Studies in Computational Intelligence, vol. 138. Springer, Heidelberg (2008)
25. Zadeh, L.A.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–127 (1997)
26. Andrysek, J., Bodini, A., Karny, M., Kracik, J., Ruggeri, F.: On combining partial and incompatible information in e-negotiaion and e-arbitration. Group Decision and Negotiation 10, 1–10 (2007)
27. Pedrycz, W.: Collaborative fuzzy clustering. Pattern Recogn. Lett. 23(14), 1675–1686 (2002)
28. Wilks, S.S.: Mathematical Statistics. Wiley Publications in Statistics. John Wiley, New York (1962)
29. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
30. Schulkin, J.: Allostasis, Homeostasis, and the Costs of Physiological Adaptation. Hardcover (2004)
31. Tapscott, D., Williams, A.D.: Wikinomics: How Mass Collaboration Changes Everything. Penguin Books Ltd., Lodon (2007)
32. Turrigiano, G.G., Nelson, S.B.: Hebb and homeostasis in neuronal plasticity. Current Opinion in Neurobiology 10, 358–364 (2000)
33. Apolloni, B., Bassis, S., Gaito, S.: Fitting opportunistic networks data with a Pareto distribution. In: Apolloni, B., Howlett, R.J., Jain, L. (eds.) KES 2007, Part III. LNCS, vol. 4694, pp. 812–820. Springer, Heidelberg (2007)
34. Parisi, G.: Brownian motion. Nature 433, 221 (2005)
35. Dahiyata, E.A.R.: Intelligent agents and intentionality: Should we begin to think outside the box? Computer Law & Security Report 22(6), 472–480 (2006)
36. Lamperti, J.: Stochastic processes: a survey of the mathematical theory. Springer, New York (1977)
37. Milner, R.: Communicating and Mobile Systems: the $\pi$-Calculus. Cambridge University Press, Cambridge (1999)
38. Cardelli, L.: From Processes to ODEs by Chemistry. In: Fifth IFIP International Conference on Theoretical Computer Science, TCS 2008, pp. 261–281 (2008)
39. Amos, M.: Theoretical and Experimental DNA Computation. Springer, Heidelberg (2005)
40. Dorigo, M., Stutzle, T.: Ant Colony Optimization. Bradford Books, The MIT Press (2004)

41. Eberhart, R.C., Shi, Y., Kennedy, J.: Swarm Intelligence. The Morgan Kaufmann Series in Artificial Intelligence. Hardcover (2001)
42. Turrigiano, G.G., Nelson, S.B.: Homeostatic plasticity in the developing nervous system. Nature Reviews Neuroscience 5, 97–107 (2004)
43. Capasso, V.: Mathematical Structures of Epidemic Systems. Springer, New York (1992)
44. Prikrylova, D., J'ilek, M., Waniewski, J.: Mathematical Modeling of the Immune Response. CRC Press, Boca Raton (1992)
45. Perelson, A.S.: Modelling viral and immune system dynamics. Nature Rev. Immunol. 2, 28–36 (2002)
46. Winans, S.: Bacterial speech bubbles. Nature, 437–330 (2005)
47. Blackwell, D., Girshick, M.A.: Theory of Games and Statistical Decisions. Dover Publications, Inc., New York (1979)
48. Apolloni, B., Bassis, S., Gaito, S., Malchiodi, D., Zoppis, I.: Controlling the losing probability in a monotone game. Information Sciences 176, 1395–1416 (2006)
49. Apolloni, B., Bassis, S., Gaito, S., Malchiodi, D.: Cooperative games in a stochastic environment. In: Apolloni, B., Kurfess, F. (eds.) From synapses to rules. Discovering symbolic rules from neural processed data, ch. 4, pp. 75–86. Kluwer Academic/Plenum Publishers, New York (2002)
50. Apolloni, B., Bassis, S., Gaito, S., Malchiodi, D.: Elementary team strategies in a monotone game. Nonlinear Analysis: Special Issues on Hybrid Systems and Applications 64(2), 310–328 (2005)
51. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)
52. Hebb, D.O.: The organization of behavior. Wiley, New York (1949)
53. Hinton, G.E., Sejnowski, T.J.: Learning and relearning in Boltzmann machines. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, pp. 282–317. MIT Press, Cambridge (1986)
54. Turrigiano, K.K., Leslie, G.G., Desai, N.S., Rutherford, L.C., Nelson, S.B.: Activity-dependent scaling of quantal amplitude in neocortical neurons. Nature 391, 892–895 (1998)
55. Desai, N.S., Rutherford, L.C., Turrigiano, G.G.: Plasticity in the intrinsic excitability of cortical pyramidal neurons. Natural Neuroscience 2, 515–520 (1999)
56. Agnati, L.F., Zoli, M., Stromberg, I., Fuxe, K.: Intercellular communication in the brain: wiring versus volume transmission. Neuroscience 69(3), 711–726 (1995)
57. Mashburn, L.M., Jett, A.M., Akins, D.R., Whiteley, M.: Staphylococcus aureus serves as an iron source for pseudomonas aeruginosa during in vivo coculture. Journal of Bacteriology 187(2), 554–566 (2005)
58. Mashburn, L.M., Whiteley, M.: Membrane vesicles traffic signals and facilitate group activities in a prokaryote. Nature 437(7057), 422–425 (2005)
59. Stewart, E.J., Madden, R., Paul, G., Taddei, F.: Aging and death in an organism that reproduces by morphologically symmetric division. PLoS Biology 3(2), 295–300 (2005)
60. Kohanski, M.A., Collins, J.: Rewiring bacteria, two components at a time. Cell 133, 947–948 (2008)
61. Skerker, J.M., Perchuk, B.S., Siryapom, A., Lubin, E.A., Ashenberg, O., Goullan, M., Laub, M.T.: Rewiring the specificity of two component signal transduction systems. Cell 133, 1043–1054 (2008)
62. Castiglione, F.: A network of cellular automata for the simulation of the immune system. International Journal Morden Physics C 10, 677–686 (1999)
63. Ferber, J.: Multi-Agents Systems. Addison-Wesley, Reading (1999)

64. McCulloch, W.S., Pitts, W.: A logical calculus of ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943)
65. Milner, R.: Functions as processes. Mathematical Structures in Computer Science 2, 119–141 (1992)
66. Phillips, A., Cardelli, L.: A correct abstract machine for the stochastic pi-calculus. In: Bioconcur 2004. ENTCS (August 2004)
67. Apolloni, B., Bassis, S., Clivio, A., Gaito, S., Malchiodi, D.: Modeling individual's aging within a bacterial population using a pi-calculus paradigm. Natural Computing 6(1), 33–53 (2007)
68. Tesfatsion, L.: Agent-based computational modeling and macroeconomics. In: Colander, D. (ed.) Post-Walrasian Macroeconomics: Beyond the Dynamic Stochastic General Equilibrium Model, Cambridge University Press, Cambridge (2006)
69. Rodriguez, A., Whitson, J., Granger, R.: Derivation and analysis of basic computational operations of thalamocortical circuits. Journal of cognitive neuroscience 16(5), 856–877 (2004)
70. Biella, G.E., Groppetti, A., Novelli, A., Fernandez-Sanchez, M.T., Manfredi, B., Sotgiu, M.L.: Neuronal sensitization and its behavioral correlates in a rat model of neuropathy are prevented by a cyclic analog of orphenadrine. Journal Neurotrauma 20(6), 593–601 (2003)
71. Izhikevich, E.M.: Simple model of spiking neurons. IEEE Transactions on Neural Networks 14, 1569–1572 (2003)
72. Apolloni, B., Bassis, B.: A feed-forward neural logic based on synaptic and volume transmission. Brain Research Reviews 55(1), 108–118 (2007)
73. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of human mobility on the design of opportunistic forwarding algorithms. In: Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM), Barcelona, Spain (2006)
74. Grossglauser, M., Tse, D.: Mobility increases the capacity of ad hoc wireless networks. IEEE Transaction on Networking 10(4), 477–486 (2002)
75. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. Wireless Networks 8(2/3), 153–167 (2002)
76. Abbott, L.F.: Lapique's introduction of the integrate-and-fire model neuron. Brain Research Bulletin 50(5/6), 303–304 (1907)
77. Hodgkin, A., Huxley, A.: A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal of Physiology 117, 500–544 (1952)
78. Zhang, J., Wang, H.: A Pi-calculus-Based Business Process Formal Design Method. In: Shen, W., Luo, J.-Z., Lin, Z., Barthès, J.-P.A., Hao, Q. (eds.) CSCWD. LNCS, vol. 4402, pp. 347–356. Springer, Heidelberg (2007)
79. Puhlmann, F.: Soundness Verification of Business Processes Specified in the Pi-Calculus. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 6–23. Springer, Heidelberg (2007)
80. Pautasso, C., Alonso, G.: Visual composition of web services. In: Proceedings of IEEE Symposium on Human Centric Computing Languages and Environments, 2003, pp. 92–99 (2003)
81. Matthewson, J.: Toward a human-centric web. ComputerUser.com, pp. 1–3 (2001)
82. Phillips, A., Cardelli, L.: Efficient, correct simulation of biological processes in the stochastic pi-calculus. In: Calder, M., Gilmore, S. (eds.) CMSB 2007. LNCS (LNBI), vol. 4695, pp. 184–199. Springer, Heidelberg (2007)
83. Zurlo, F., Apolloni, G., Castelli, A.: The "everyday feature" of designing. the project in terms of a new social capital. Re-Public (July 2008)

# Autonomous Composition of Fuzzy Granules in Ambient Intelligence Scenarios

Giovanni Acampora[1], Vincenzo Loia[1], and Athanasios V. Vasilakos[2]

[1] Department of Mathematics and Computer Science, University of Salerno, Italy
[2] Department of Computer and Telecommunications Engineering,
University of Western Macedonia, Greece

**Abstract.** Pervasive and human-centric computing is beginning to be fact: with cell phones, laptops and handhelds, human beings can work pretty much anywhere. Ambient Intelligence (AmI) is a novel human-centric computer discipline based on three emergent technologies: Ubiquitous Computing, Ubiquitous Communication and Intelligent User Interfaces. The integration of aforesaid technologies opens new scenarios for improving the interaction between humans and information technology equipments realizing a human-centric computing environment. Within this aim the deliverable of tasks or services should be achieved through the usage of an invisible network of heterogeneous devices composing dynamic computational-ecosystems capable of satisfying the users requirements. Fuzzy granules, intended as a clump of objects which are drawn together by criteria like indistinguishability, similarity, proximity or functionality, can represent a powerful and, simultaneously, simple paradigm to embed intelligence into a generic AmI ecosystem in order to support people in carrying out their everyday life activities, tasks and rituals in easy and natural way. However, the strong dinamicity and the high complexity characterizing a typical AmI scenario make difficult and expensive to design *ad-hoc* fuzzy granules. This paper presents a framework exploiting methodologies coming from Semantic Web and Computational Intelligence areas to compose fuzzy granules in autonomous way in order to maximize the users comfort and achieve the hardware transparency and interoperability.

**Keywords:** Ambient Intelligence, Granular Computing, User-Centric Systems, Information Retrieval, Fuzzy Logic, Fuzzy Control, Autonomous Systems, Adaptive Systems, Semantic Web, Home Automation, Network Computing, Evolutionary Methodologies, Markup Languages, Ubiquitous Computing, Ubiquitous Networking, Advanced Intelligent User-Friendly Interfaces, Multi-Agent Systems, Intelligent Agents, Cooperative Agents, Adaptive Fuzzy Systems.

## 1 Introduction

Futurists tend to agree that personal computing will be dramatically changed in the future. One overriding objective will be to make the technology transparent to the user, thus eliminating the frustration that many users face today. Human-Centric computing systems are pervasive frameworks capable of creating a solution so that the human is always connected, portable, and available. In this context, AmI systems represent one of the most emergent technologies able to offer advanced user-oriented services. Indeed, AmI systems will radically change how people interact with technology: the principle is to integrate different computer science backgrounds with psychology and social sciences in order to create a network of intelligent devices (sensors and actuators) able to enhance the quality of people's life [23]. This is possible thanks to systems' ability in

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 265–287.
springerlink.com

anticipating needs and desires necessary to obtain safety, comfort and economy. AmI systems realize such requirements using the following design philosophies:

- Context Awareness;
- Multi-modal Communication;
- User-Centered interaction.

According to a more formal definition of Context Awareness [17], we can say that contextual information can be defined as an ordered multilevel set of declarative information concerning events occurring both within the sensing domain of a smart space and within the communication and action domains of the smart space itself. In particular, an event can be defined as the occurrence of some facts that can be perceived by or communicated to the ambient intelligence environment. Different attributes can characterize an event: where (location), what (core), when (time), why (reason). Multimodal communication can then be defined as the simultaneous exchange of information over multiple channels at different levels of abstraction [15]. Human-to-human communication is intrinsically multi-modal. Multimodal interactions in AmI systems allow the artificial system to engage in a similar dialog with the user, with the objective of exploiting the richness, robustness, and flexibility of face-to-face conversation. Designing architecture to support user-centered interactions requires a high degree of flexibility and adaptation. In a user-centered paradigm it is the system that tries to meet the user's personal interaction style and not vice-versa as often happens.

By analyzing the aforesaid AmI features, it can be asserted that homes, or more generally buildings, are changing their nature from static structures of bricks to dynamic work and living environments that actively support and assist their inhabitants [19]. These novel living environments are expected to behave in intelligent way. In addition, to satisfying the needs of its inhabitants, a building has to be an active, autonomous entity that pursues its own goals (energy consumption, security, etc.). To fulfill this goal, a smart home must continually take decisions by specifying rules that describe which actions to take.

Recently, different Computational Intelligence methodologies have been exploited to define smart rules able to control AmI devices in autonomous way in order to optimize several living parameters [14][24].

Beyond purely computational issues, the design and implementation of intelligent living environments are highly influenced from the hardware infrastructure exploited to model the *control network* interconnecting the collection of sensors/actuators composing the AmI system. Today, different communication protocols [25] [26] [10] could be used to realize an intelligent environment and, consequently, several programming acquaintances are necessary to deal with these protocols. This paper focuses on the integration of methodologies coming from different computer science areas as the Computational Intelligence and Semantic Web to embed intelligence into the AmI devices and define an abstract developing environment capable of dealing with different hardware protocols by means of granular computing and semantic web technologies to attempt to solve both AmI issues.

Fuzzy Granular Computing is a theory dealing with the partitioning of a class of objects into granules, with a granule being a clump of objects, which are drawn together by indistinguishability, similarity or functionality. In our case a fuzzy granule is an intelligent entity capable of controlling a portion of a living environment by means of a clump of objects, where each object deals with a well-defined subset of sensors/actuators. Some samples of granules composing an AmI environment are: the *temperature control granule*, the *lighting control granule*, the *blind control granule*, and so on. Granular computing in AmI allows collecting control entities with the same functionalities in homogenous groups in order to maximize the cohesion level of the AmI system components; the cohesion is one of the most important factors characterizing the Software Engineering discipline. The maximization of system cohesion allows designer to focus on well-defined part of system and the interface among them guaranteeing a satisfactory engineering process. The fuzzy granules are suitable logical model to represent intelligence and achieve the cohesion property.

However, the strong dynamicity and the high complexity characterizing a typical AmI scenario make difficult and expensive the design of *ad-hoc* fuzzy granules to control AmI devices. Our work realizes an AmI framework capable of composing the appropriate fuzzy granules in autonomous way and performing their services in hardware-independent way.

## 2 A Framework for AmI Autonomous Fuzzy Granular Computing

This section explains the architecture of the proposed AmI system through the presentation of the theoretical concepts and technologies used to design and realize the system.

### 2.1 AmI Autonomous Fuzzy Granular Computing Architecture

A ubiquitous computing fuzzy system exploits fuzzy computation in living environments in order to enable people to move around and interact with their environment more naturally than they actually do. More precisely, a ubiquitous computing fuzzy system is defined as a network of devices able to regulate their behavior in an automatic way according to user needs and preferences. In order to achieve this goal different research topics have to be considered as *computational intelligence, distributed computing* and *sensor networks*.

Computational intelligence approaches model the environmental context and relationships among the events occurring in AmI scenarios, whereas, distributed computing and sensor network technology are required to model the real physical AmI environment and allow the communication among devices composing the framework. The joint use of these techniques allows achieving a fully automated and optimized control of environment according to user's preferences.

Starting from the integration among aforesaid topics, the proposed AmI scenario can be viewed as a composition of intelligent entities, named granules, each one composed of a collection of fuzzy controllers managing devices characterized from the same functionality as, for instance:

- Lighting granule;
- HVAC granule;
- Blinds granule;
- Temperature granule;
- …

where the lighting granule is the collection of control objects able to manage each light actuator in the framework, and so on.

In other word each granule is a collection of fuzzy functions able to deal the same kind of device. Figure 1 shows a typical AmI granular scenario.

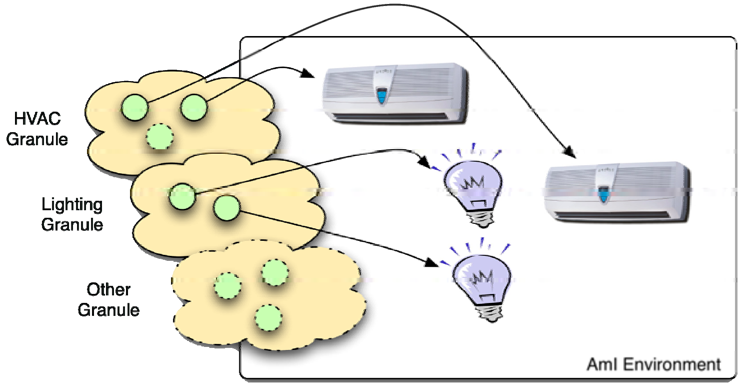


**Fig. 1.** Ambient Intelligence Granular Scenario

However, the high dynamicity and complexity characterizing an AmI scenario make very difficult and expensive to design *ad-hoc* fuzzy granules to control AmI devices. The dynamicity and complexity of AmI environments arise from two factors: the amount and type of devices populating the environment and the hardware heterogeneity.

Aim of this paper is to realize an AmI environment capable of composing the appropriate fuzzy granules in autonomous way and performing their services in hardware-independent way.

The proposed features are achieved by considering the system as a Distributed Service Oriented Architecture, where each service corresponds to a control object of a fuzzy granule.

In order to achieve the autonomous granules composition and the hardware independence a double representation of granule objects is required: a *Fuzzy Markup Language* (FML) representation allows to manage the hardware independency, whereas, a *Resource Description Framework* (RDF) model is used as objects description for the autonomous granules composition algorithm. Java, Web Services and JAXB technologies allow running the FML/RDF granule objects.

The overall view of our architecture is modeled in terms of four fundamental and complementary sub-systems whose interactions allow designing, developing and putting in work the AmI fuzzy granules. The four sub-systems are:

- *Granular Design Environment*;
- *Granular Run Time Environment*;
- *Granular Service Retrieval Environment*;
- *AmI Environment.*

The *Granular Design Environment* subsystem is an FML-based framework modeling and managing the fuzzy granule in a hardware-independent and human-oriented way.

The *Granular Run Time Environment* subsystem is a Java-based framework able to compute the AmI granular objects through the integration of Web Services and JAXB technologies.

Granular control activities are managed as services, modeled by FML and translated into RDF representation in order to be indexed by the *Granular Service Retrieval Environment* subsystem.

*AmI Environment* subsystem defines the set of fuzzy controlled devices composing the sensor network.

The communication among the Run Time, Service Retrieval and the AmI environment is accomplished by the Internet protocol suites (TCP/IP) and in particular the HTTP application protocol. From the communication point of view, the ubiquitous fuzzy computing framework may be considered as a Client/Server system where the clients are located in the AmI environment and the servers are hosted in RunTime and Service Retrieval environments.

Within this scenario, we distinguish three basic entities:

1. *AmI Client*. This entity is located in Fuzzy Control AmI environment; the AmI Client demands the appropriate fuzzy objects to Retrieval Server in order to compose AmI fuzzy granules;
2. *Retrieval Server*. It hosts the fuzzy objects and it performs the retrieval fuzzy algorithm used by AmI Client to compose the control granules;
3. *Run Time Server*. It computes the remote granular fuzzy object.

Figure 2 shows a high-level view of system architecture with AmI Clients, Retrieval Server and Run Time Server. Before to introduce the framework components in a detailed fashion, a survey about FML, the fuzzy objects description language, will be given.

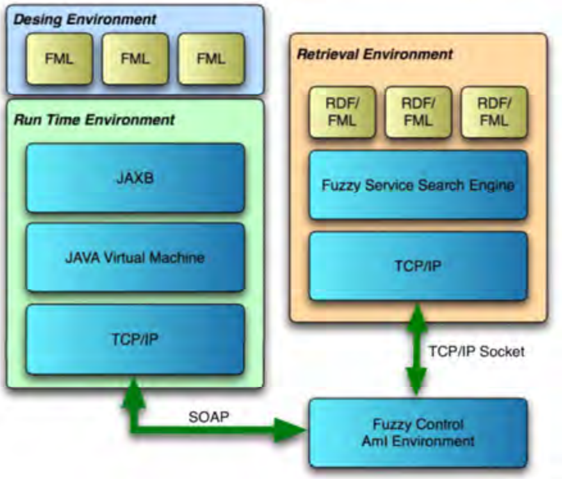


**Fig. 2.** AmI Granular Ubiquitous Fuzzy System

## 3 Merging Fuzzy Logic and XML: The Fuzzy Markup Language

In section 1 the AmI context has been defined as an ordered multilevel set of declarative information concerning events occurring both within the sensing domain of a smart space and within the communication and action domains of the smart space itself. Each event consists of a set of features, nominally, when the event has been generated, what generated the event and in which context the event has been generated. In order to model this information it is necessary to use theoretical concepts representing the different AmI context events in an optimal and natural way. Fuzzy Logic and, in particular, the rules of a fuzzy controller can represent a direct translation of AmI context definition. In fact, taking into account a generic fuzzy rule it is possible to derive each event-related attribute of context definition (what, when, why, etc.) as shown in figure 3. Moreover, the AmI environment can be viewed as a collection of entities that are measurable in fuzzy way. For instance is very simple to model the environmental temperature notion by means of a fuzzy concept containing three fuzzy sets labeled, respectively, LOW, MEDIUM and HIGH and each one mapped on an opportune fuzzy support; the shape of these sets can be chosen in a static way or through learning mechanism as, for instance, evolutionary methods. From this point of view, fuzzy logic offers several benefits related to the modeling of complex systems and the straightforwardness of embedding fuzzy controllers in advanced evolutionary approaches in a rapid way.

Furthermore, fuzzy control theory allows defining the relationships among AmI information in a linguistic way, i.e., by using the same idea of a human being which wants to regulate the living environment in order to satisfy its need and requirements [8]; from a scientific point of view, fuzzy controllers simplify the design and development of automatic mechanisms to self-regulation of AmI entities, in fact, the linguistic approach results, remarkably, more fast and direct than classic PID design methods.

FML is a markup-based general approach to modeling the fuzzy objects and the set of relations within an AmI environment by using a human-oriented and a hardware-independent syntax. Our approach uses the FML description to define object collections, each one able to control a well-defined kind of device; these collections represent the AmI fuzzy granules.

Details of FML and how FML can be incorporated into the Design subsystem of our AmI framework are found in the following subsection.

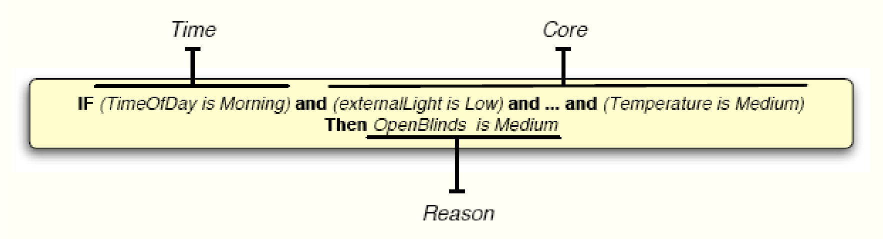

Time
Core
IF (TimeOfDay is Morning) and (externalLight is Low) and ... and (Temperature is Medium)
Then OpenBlinds is Medium
Reason

**Fig. 3.** AmI Fuzzy Control Rule

### 3.1 Transparent Fuzzy Control for AmI Context Representation

This section is devoted to present FML, the main tool exploited to design fuzzy object composing AmI granules.

From a technological point of view, fuzzy control deals with the controller implementation on a specific hardware by using a public or legacy programming language. For this reason, independently from the complexity of the addressed application problem, the development time may be very expensive. In Ambient Intelligence (AmI) environments, where the ubiquitous computing represents one of the main features, the possibility of dealing with a considerable number of heterogeneous controlled hardware is high enough to constitute a real impediment to a flexible and efficient control strategy. In order to solve this drawback, software layers, designed to control hardware details, are extremely useful. FML (Fuzzy Markup Language) is a software technology to create fuzzy oriented abstraction tools. FML is XML-based and its syntax is realized by using a set of tags representing the different components of a fuzzy controller.

Since Zadeh's coining of the term fuzzy logic [27] and Mamdani's early demonstration of Fuzzy Logic Control (FLC) [12], the scientific community in the theoretical as well as the application fields of FLC has made enormous progress. A fuzzy control allows the designer to specify the control in terms of sentences rather than equations by replacing a conventional controller, say, a PID (proportional integral-derivative) controller with linguistic IF-THEN rules [13]. As described in previous sections, the main components of a fuzzy controller are:

- Fuzzy Knowledge Base;
- Fuzzy Rule Base;
- Inference Engine;
- Fuzzification sub-system;
- Defuzzification sub-system.

The Fuzzy Knowledge Base contains the knowledge used by human experts. The Fuzzy Rule Base represents the set of relations among fuzzy variable defined in the controller system. The Inference Engine is the fuzzy controller component able to extract new knowledge from a fuzzy knowledge base and a fuzzy rule base. Extensible Markup Language (XML) [22] is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, nowadays XML plays a fundamental role in the exchange of a wide variety of data on the Web, allowing designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications, devices and organizations. If we use XML, we take control and responsibility for our information, instead of abdicating such control to product vendors. This is the motivation under FML proposal: to free control strategy from the device. The technologies used in FML are:

- XML in order to create a new markup language for FLC;
- XML Schema in order to define the legal building blocks of an XML document.

Initially, FML relied on XML Document Type Definition (DTD) [3] because this approach is able to translate in a direct and simple way the context free grammar theoretical concepts into a usable markup language speeding up the language definition.

More recently [2], FML has been defined by using XML Schema. The set of data types composing a fuzzy controller model using the FML language is structured as an n-ary tree called FOM (Fuzzy Objects Model). Reasoning in this way, it is possible to state that each FML program can be associated to an instance of a FOM tree. A portion of XML Schema generating the FML syntax is shown in listing 1.

Currently, we are using FML for modeling two well-known fuzzy controllers: Mamdani and Takagi-Sugeno-Kang (TSK) [21]. In order to model the Controller node of a fuzzy tree, the FML tag *<FuzzyController>* is created (this tag opens any FML program). *<FuzzyController>* uses three tags: *type*, *defuzzificationMethod* and *ip*. The type attribute allows to specify the kind of fuzzy controller, in our case Mamdani or TSK; defuzzificationMethod attribute defines the defuzzification method used to translate the fuzzy results coming from fuzzy inference engine application into real double system control values; ip tag will be defined at the end of section. Considering the left sub-tree, the knowledge base component is encountered. The fuzzy knowledge base is defined by means of the tag *<KnowledgeBase>* that maintains the set of fuzzy concepts used to model the fuzzy control system. In order to model each fuzzy concept belong in fuzzy knowledge base, it is necessary to use the following XML elements:

- *<FuzzyVariable>*;
- *<FuzzyTerm>*;
- a set of tags used to model the shapes defining the fuzzy variable;

*<FuzzyVariable>* defines the single fuzzy concept, for example *Luminosity*; *<FuzzyTerm>* defines a linguistic term describing the fuzzy concept, for example low (luminosity); the set of tags defining the shapes of fuzzy sets are related to fuzzy terms. The attributes of *<FuzzyVariable>* tags are: *name*, *scale*, *domainLeft*, *domainRight*, *type*, *ip*. The name attribute defines the name of fuzzy concept (i.e. time of the day); scale defines how to measure the fuzzy concept (i.e. hour); domainLeft and domainRight model the universe of discourse of fuzzy concept in terms of real values (i.e. [0000, 2400]); the role of variable (i.e. independent or dependent variable) is defined by type attribute; ip locates the position of fuzzy knowledge base in the computer network. *<RuleBase>* allows the building of the rule base associated with the fuzzy controller. This tag uses the following attribute: ip. The other tags related to rule base definition are:

- *<Rule>*;
- *<Antecedent>*;
- *<Consequent>*;
- *<Clause>*;
- *<Variable>*;
- *<Term>*;
- *<TSKParams>*;
- *<TSKParam>*.

The *<Rule>* tag defines a single fuzzy rule by using the *<Antecedent>* and *<Consequent>* nested tags; both tags model the fuzzy propositions appearing, respectively, in antecedent and consequent part of a single rule. Each antecedent fuzzy proposition is modeled by means of *<Clause>* tag and its nested elements: *<Variable>* and *<Term>*.

Analogously, each consequent fuzzy proposition is defined by means of <*Variable*> and <*Term*>, in the case of Mamdani controller, or by means of <*Variable*>, <*TSKParams*> and <*TSKParam*>, in the case of Takagi-Sugeno-Kang controller.

```
<?xml v e r s i o n=1.0 encoding =UTF 8>
<!edited with XMLSpy v2005
sp1U (http://www. xmlspy.com)
by Gianni Acampora (University of  Salerno)
>
<xs:schema xmlns:xs=http://www.w3 .org/2001/XMLSchema
    elementFormDefault=qualified
    attributeFormDefault=unqualified >
<xs:element
    name = FuzzyController
    type = FuzzyControllerType>
    <xs:annotation>
        <xs:documentation>
            FuzzyControllerMarkupProgram
        </xs:documentation>
    </xs:annotation>
</xs:element>
<xs:complexType name = FuzzyControllerType>
    <xs:sequence>
        <xs:element
            name = KnowledgeBase
            type = KnowledgeBaseType>
            <xs:annotation>
                <xs:documentation>
                    Fuzzy Concepts Collection
                </xs:documentation>
            </xs:annotation>
        </xs : element>
        <xs : element name = RuleBase type = RuleBaseType>
            <xs:annotation>
                <xs:documentation>
                    FuzzyRulesCollection
                </xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
    <xs:attribute name = ControllerType>
        <xs:simpleType>
            <xs:restriction base = xs:string >
                <xs:pattern value = mamdani | tsk/>
            </xs:restriction >
        </xs:simpleType>
    </xs:attribute>
. . .
```

**Listing 1.** Fuzzy Markup Language XML Schema

Differently from other attributes used in FML language, the ip attribute is not directly related to the fuzzy logic controller theory. In fact, this attribute contains information defined by means of the following regular expression (expressed in XML Schema syntax):

(1?[0−9]?[0−9]|2[0−4][0−9]|25[0−5]).)3(1?[0−9]?[0−9]|2[0−4][0−9]|25[0−5] :?._)

It is simple to see how this regular expression defines strings such as:

- 192.168.0.4;
- 192.168.0.4:8080;
- 192.168.0.4.8080/FMLWebService;
- etc.

Hence, ip attribute represents TCP/IP endpoint; for instance, in [4] the ip attribute of *<FuzzyController>* tag is used to define the address of a TCP Berkeley Socket based Server computing FML controllers generated in automatic way through a fuzzy inductive algorithm, whereas, in [1] the ip attribute of *<FuzzyVariable>* and *<Rule>* tags is used to distribute (in order to minimize the inference time) the different part of controller on the network by means of a multi-agent system. In this paper the ip attribute will be used to define the endpoint of web service computing the FML controller.

Listing 2 gives a sample of FML code.

```
<!DOCTYPE FUZZYCONTROL SYSTEM ”fml . dtd”>
<FUZZYCONTROL de f uz z i f yme thod = ”CENTROID” ip = ”localhost”
        type = ”MAMDANI”>
<KNOWLEDGEBASE IP = ”localhost”>
<FUZZYVARIABLE
        domainleft = “0” domainright = “1”
        ip = ”localhost” name = ”Luminosity”
        scale = ”Lux” type = ”INPUT”>
        <FUZZYTERM name=”low”>
                <PISHAPE
                        param1 = ”0.0”
                        param2 = ”0.45”>
                </PISHAPE>
        </FUZZYTERM>
        <FUZZYTERM name=”MEDIUM”>
                <PISHAPE
                        param1 = ”0.49999999999999994”
                        param2 = ”0.44999999999999996”>
                </PISHAPE>
        </FUZZYTERM>
        <FUZZYTERM name=”HIGH”>
                        <PISHAPE
                                param1 = ”0.5501”
                                param2 = ”1”>
                        </PISHAPE>
                </FUZZYTERM>
</FUZZYVARIABLE>
</KNOWLEDGEBASE>
```

**Listing 2.** FML sample program

```
<RULEBASE
    inferenceengine = ”MINMAXMINMAMDANI”
    ip = ”localhost”>
    <RULE connector = ”AND” ip = ”localhost” weight = ”1”>
        <ANTECEDENT>
            <CLAUSE not = ”FALSE”>
                <VARIABLE> Luminosity </VARIABLE>
                <TERM> low </TERM>
            </CLAUSE>
            <CLAUSE not = ”FALSE”>
                <VARIABLE> hour </VARIABLE>
                <TERM> morning </TERM>
            </CLAUSE>
        </ANTECEDENT>
        <CONSEQUENT>
            <CLAUSE not = ”FALSE”>
                <VARIABLE>dimmer</VARIABLE>
                <TERM>medium</TERM>
            </CLAUSE>
        </CONSEQUENT>
    </RULE>
. . .
</RULEBASE>
</FUZZYCONTROL>
```

**Listing 2.** (*continued*)

## 4 Run Time Subsystem: Implementing the FML Fuzzy Objects

The FML codes represent only a human-oriented and hardware-independent representation of a fuzzy granule objects, i.e., the FML granules cannot be computed in a direct way. In other words, an FML compiler is needed to translate the FML model representing the fuzzy granules into an executable program. We explored different approaches to implementing the FML compiler: XSLT Stylesheet Translator, JAVA XML Parser (JAXP) or other XML-based translator technologies. The results led to the current implementation based on the integration of JAXB (Java Architecture for XML Binding) with TCP/IP suites protocol.

JAXB represents a direct way to compile and compute the FML services. In fact, the JAXB allows translating the XML tree structure (in our case, the FOM) into a Java class’s hierarchy in a direct and simple way via the *xjc* compiler. The TCP/IP stack allows the design and the development of a remote FML controller; in particular, the proposed system uses SOAP protocol together with web-services technologies in order to remote the control task. Specifically, JAXB can generate Java classes from XML schemas by means of a JAXB binding compiler. The JAXB binding compiler takes XML schemas as input, and then generates a package of Java classes and interfaces, which reflect the rules defined in the source schema. These generated classes and interfaces are in turn compiled and combined with a set of common JAXB utility packages

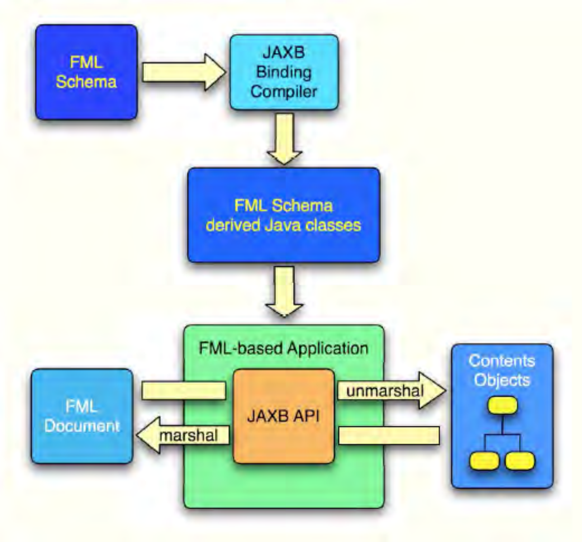


**Fig. 4.** The JAXB/FML/Java binding

to provide a JAXB binding framework. The JAXB binding framework provides methods for unmarshalling XML instance documents into Java content trees, a hierarchy of Java data objects that represent the source XML data, and for marshalling Java content trees back into XML in-stance documents. The JAXB binding framework also provides methods for validating XML content as it is unmarshalled and marshalled. A JAXB compiler uses the XML Schema related to FML to build the class hierarchy, and a set of API to unmarshal the FML file into fuzzy objects hierarchy. The JAXB/FML/Java binding is depicted in figure 4. The generated objects hierarchy represents only a static view of FML file. This resource does not embody the fuzzy methods/operators necessary to perform deduction activity over the fuzzy structures. In order to complete the Java representation of FML fuzzy controllers, a fuzzy wrapper class, named *FMLController* has been coded. In particular, the *FMLController* class exhibits a set of methods able to apply the appropriate fuzzy operators to the information derived from JAXB objects. Specifically, *FMLController* constructors allow the creation of a new fuzzy controller by using the unmarshall method of JAXB-API independently from the FML file location (file system or network). Moreover, the public method named inference applies the opportune deduction engine to the fuzzy information contained in JAXB objects. The signature of the inference method is: *double inference(double[] input)*. The inference method reads double values from the controlled system, applies: the fuzzification operator and the inference engine in sequence, the defuzzification operator and, finally, returns a double value to the system.

The interaction between the controlled system and the *FMLController* class is performed by two abstract methods, *double[] readInputs()* and *double writeOutput(double)* whose implementation depends upon network protocol used to interface the controlled system with AmI Client.

### 4.1 Granular Fuzzy Remote Control

The granules composing an AmI environment represent only a logical entity capable maximizing typical software engineering attributes, as the cohesion, but, however, the granules objects are distributed on a computer network as FML program and computed through Java/JAXB technologies. In other word the granules objects are distributed objects. In order to perform a remote execution of FML granular objects, it is necessary to embed the granular runtime framework into a Web Services environment. According to the W3C, a Web Service [5] is a software system designed to support interoperable machine-to-machine interaction over a computer network. It has an interface that is described in a machine-readable format such as Web Services Description Language (WSDL). Other systems interact with the Web service in a manner prescribed by its interface using messages, which may be enclosed in a Simple Object Access Protocol (SOAP) envelope. These messages are typically conveyed using HTTP protocol and normally comprise XML in conjunction with other Web-related standards. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to interprocess communication on a single computer. This interoperability (for example, between Java and Python, or Microsoft Windows and Linux applications) is due to the use of open standards. The intrinsic interoperability offered by the web services communication paradigm covers in a direct way the interoperability concepts required by the ubiquitous properties of an AmI system. From this point of view the web services represent the communication core of the proposed AmI system. In fact, it is this interoperability of the web services property that is fundamental to achieving the ubiquitous computing and ubiquitous networking properties of AmI systems.

In order to compute our FML controller through Web Services technology it is necessary to use a specific Web Services engine. Our framework uses the Axis engine to deploy the inference service by means of a Deployment Descriptor (WSDD) format. Once deployed, the service is ready to accomplish its work when invoked by clients. Obviously, the clients have to know the address of web services procedure, i.e., the web services endpoint. The IP attribute of *<FuzzyController>* tag present in FML captures the endpoint information.

## 5 Retrieval Subsystem: A Semantic View of Fuzzy Granules

This section represents the core of proposed system. It is devoted to present a methodology able to retrieve fuzzy object from the computer networks in order to compose fuzzy control granules. In details, the proposed algorithm is able to find the most suitable set of controllers for a prefixed AmI environment by exploiting a semantic representation of FML controllers and the ip attributes of FML programs.

The ip attribute of the root tag of the FML program has been introduced as a key element to store the web services endpoint address. We present an approach, based on Semantic Web technology, suitable to retrieving the appropriate FML endpoint. The basic idea is to exploit information arising from the set of sensor/actuator devices composing the AmI environment. The endpoint search engine is located on a Retrieval Server (see section 2) and it can be viewed in terms of three components:

- AmI Sensor Network Knowledge Client;
- Storing Algorithm;
- Retrieval Algorithm.

The AmI Sensor Network Knowledge Client is located on the AmI clients; the Storing and Retrieval Algorithms are located on the Retrieval Server. The AmI Sensor Network Knowledge Client collects information from the environment and synthesizes it in order to trigger the appropriate FML controller request. The Storing and Retrieval algorithms are designed, respectively, to catalogue FML information in a semantic way and to manage this repository for fuzzy controller searching. The Retrieval algorithm uses information generated by Storing algorithm together with information coming from AmI clients in order to return the appropriate FML Web Services endpoint.

The repository description is based on RDF, a well-known technology coming from Semantic Web.

In the next subsection we show the fundamental steps concerning three basic activities: the collecting of AmI Sensor Network Information, repository building by using RDF and the FML endpoint search mechanism.

### 5.1 AmI Sensor Network Knowledge

The AmI clients have to model the controlled Sensor/Actuator Network in a formal way to communicate the appropriate information to the Retrieval Server. A Sensor/Actuator network is a huge network of distributed devices using sensors/actuators

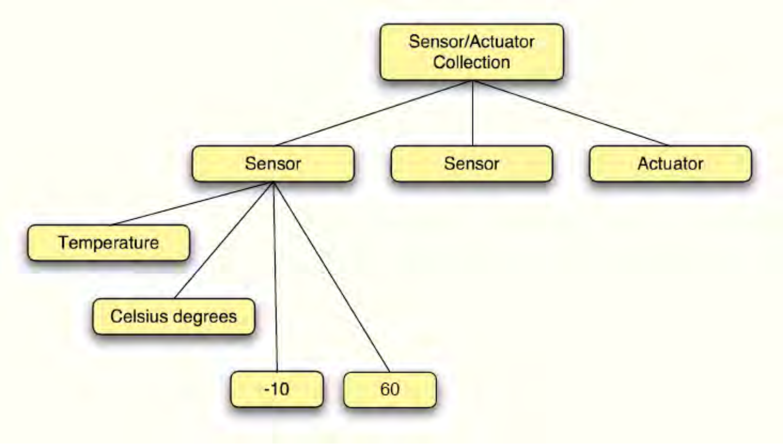


**Fig. 5.** Sensor Network Tree

to monitor/control conditions at different locations. It is possible to use a labeled tree data structure to model this network containing the following information:

- Number of devices (level one);
- Device information (level two):
  - Type (Sensor/Actuator);
  - Monitored/Controlled entity (e.g. Temperature) and its scale (e.g. Celsius degree);
  - Set of allowable value (e.g. [10, 60])

Figure 5 shows an instance of Sensor Network tree. The information contained in the labeled tree can be modeled in a machine-readable representation by using XML. This type of information modeling is used by the Retrieval algorithm to identify the most suitable FML Web Services endpoints to return to AmI Client.

### 5.2 Storing Algorithm

The Retrieval Servers have to perform a semantic storing of FML controllers hosted on it by using the RDF technology.

RDF is a W3C recommendation [16] that was originally designed to standardize the definition and use of metadata-descriptions of Web-based resources. However, RDF is equally well suited for representing arbitrary data, be they metadata or not. The basic building block in RDF is an subject-predicate-object triple, commonly written as P(S,O). That is, a subject S has a predicate (or property) P with value O. Another way to think of this relationship is as a labeled edge between two nodes: [S] − P − [O]. This notation is useful because RDF allows subjects and objects to be interchanged. Thus, any object from one triple can play the role of a subject in another triple, which amounts to chaining two labeled edges in a graphic representation. RDF also allows a form of reification in which any RDF statement itself can be the subject or object of a triple. This means graphs can be nested as well as chained. The RDF Model and Syntax specification also proposes XML syntax for RDF data models. The FML/RDF storing mechanism is performed through two different steps:

1. FML fuzzy services semantic translation;
2. FML Semantic storing into semantic repository.

The information contained in FML files represents only a tree-structured model, tags oriented, of a fuzzy controller providing the main benefits of XML representation. However, the information modeled by an XML representation (FML, in our case) are not semantically defined, i.e., XML doesn’t allows the definition of a set of fuzzy-relations between the defined tags and attributes. For instance, by using the FML syntax, the fuzzy variables defined into knowledge base sub tree are not the same entities contained in fuzzy rules; in fact different tree nodes are used to model the same concept.

Using the RDF technology solves this drawback. The RDF metadata model is based upon the idea of making statements about resources in the form of a subject-predicate-object expression, called a triple in RDF terminology. In our case, the resources are the FML fuzzy components. In this section, the semantic representation of fuzzy controllers is introduced in terms of RDF. The semantic translation is accomplished by using XSLT where the domain is the set of fuzzy services modeled by FML and the codomain is the set of semantic models of fuzzy services expressed by RDF. The XSLT function computes a translation from the FML tree (the FOM) to the RDF graph. The resulting graph differs from the input FML tree in the number of edges composing the data structure. In fact, the RDF graph shows, differently from the FML tree, a set of graph cycles representing the fuzzy relations among the fuzzy components modeled in the FML representation. Both definitions are obtained from a FML service description by using the XSLT translation. Once he semantic representation of FML service has been obtained it is necessary to store it in a semantic repository in order to enable the users (people or external systems) to query information about the web services controller endpoint. The proposed AmI framework uses Sesame [6] server in order to manage the semantic repository and Sesame SAIL API to enable the semantic queries from external systems, for example, the AmI clients. The execution flow of the storing algorithm is shown in figure 6.

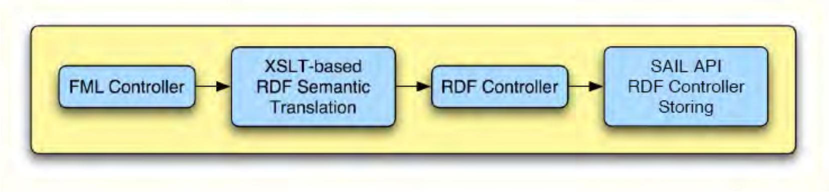

FML Controller
XSLT-based RDF Semantic Translation
RDF Controller
SAIL API RDF Controller Storing

**Fig. 6.** FML to RDF: Storing Algorithm

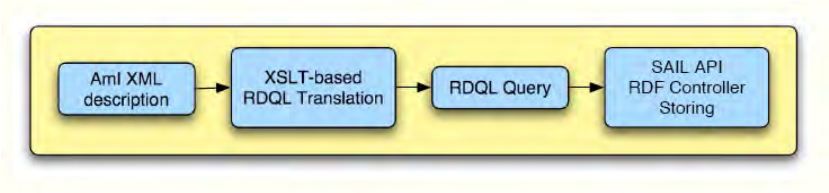


**Fig. 7.** XML to RDQL: Semantic query creation

### 5.3 Retrieval Algorithm

The Retrieval Algorithm represents the computational kernel of the AmI Granular Retrieval component. Its main aim is to find the appropriate fuzzy service able to control the environment by using the semantic information contained in the RDF description located on the Retrieval Server and the XML information coming from the AmI client. The appropriate fuzzy service is found by comparing, in a semantic way, the information contained in the XML description of the AmI environment with the triple set of RDF fuzzy service description.

RDQL [20] represents a fast and direct way to retrieve semantic information from RDF repositories. Its syntax is based on the classic SQL query language. Thus, the steps of the retrieval algorithm are:

1. to accept requests from AmI clients in XML format;
2. to convert the XML information into a string representing an RDQL query;
3. to compute the RDQL query obtained in a previous step;
4. to return the FML endpoint information to the AmI client.

The Retrieval Servers communicate with the AmI Client by using the TCP Sockets. In particular, the Retrieval Servers listen to the client requests on a prefixed TCP port, open a TCP socket communication with client, accept the XML information, compute the XML data and return the appropriate endpoint value on socket.

Once it has received the XML AmI description from the AmI client, the Retrieval server creates an XML file containing this information and uses it to compute the RDQL query. The transformation from XML to RDQL is accomplished by using the XSLT tool as shown in figure 7. In particular, the XML-RDQL XSLT translation maps each XML entry into a RDQL query portion generated by using the template of listing 3.

Starting from the XML code, representing the client knowledge, and the RDQL template presented in listing 3, the XSLT translation generates the RDQL query shown in listing 4. Row 1 represents the (eventual) return data of query computation, the FML endpoint, whereas, the query portion between row 3 and row 10 identifies the parameters of input variable.

```
SELECT ? endpoint
WHERE ?x <fml : hasEndPoint> ? endpoint
          ?x <fml : hasKnowledgeBase> ?y
          ?y <r df : li > ? z
          ? z <fml : hasName> NameOfVariable
          ? z <fml : hasScal e> ScaleOfVariable
          ? z <fml : type> TypeOfVariable
          ? z <fml : domainLeft> ? inf
          ? z <fml : domainRight> ? sup
          AND ? inf<=LeftDomainOfVariable
          ? sup>=RightDomainOfVariable
          USING fml = http://www.dmi .unisa.it/fml#
```

**Listing 3.** RDQL query template

```
SELECT ? endpoint
WHERE ?x <fml : hasEndPoint> ? endpoint
          ?x <fml : hasKnowledgeBase> ?y
          ?y <rdf : li > ? z
          ? z <fml : hasName> Temperature
          ? z <fml : hasScale> Celsius Degree
          ? z <fml : type> input
          ? z <fml : domainLeft> ? inf
          ? z <fml : domainRight> ? sup
          AND ? inf <=10 ? sup>=60
          USING fml = http://www.dmi.unisa .it/fml#
```

**Listing 4.** RDQL/XSLT query sample

This query portion is univocally defined for each XML request. The rest of the query is obtained starting from the information contained in the XML request. In particular, the row 3 is used to retrieve the Fuzzy Knowledge Base information from the RDF fuzzy model in order to analyze its components (the fuzzy variables). The analysis of each fuzzy variable is accomplished from row 4 to row 10. Particularly, rows from 5 to 9 are used to set the fuzzy variable name, scale, type and the universe of discourse coming from the AmI environment into RDQL query.

Once it has realized the RDQL query, the retrieval algorithm uses the Sesame SAIL API in order to deduct new information from the RDF repositories. In particular, this information is represented from the endpoint of the FML controller able to control the data contained in the XML request. This endpoint, successively, is sent to the AmI client where it will be used to interface the controlled AmI environment with the automatic remote fuzzy controller.

## 6 Case Study and Experimental Results

As mentioned in the paper introduction, the proposed granular ubiquitous system can be exploited in many applicative fields, but its potential is clearly highlighted by human-centric applications. For this reason a domotic application (as known as Home Automation application) could be represent an interesting case study on which the FML based ubiquitous findability system can be applied to check and validate the suitability of proposed framework to real world applications.

Each domotic application needs of control network infrastructure capable of interconnecting the set of sensors/actuators among them and interfacing this network with the typical TCP/IP data network. By means of this interface is possible to control the set of domestic devices through a network of dedicated agents hosted on distributed computing devices. Our application exploits the Lonworks technologies and, consequently, the Lontalk control network protocol to manage the details about ambient intelligence internetworking.

In order to test the proposed ambient intelligence framework, it is fundamental to simulate the upload of FML files on the servers introduced in previous sections. This simulated controllers uploading will allows FML clients to retrieve the appropriate controllers by means of semantic analysis based on RDF language. This case study exploits a modified evolutionary approach to generate, randomly, a collection of FML files that, successively, will be uploaded on aforesaid servers. At same time, the FML client hosted in our test environment will query the servers to search the right controllers. Our aim is to quantify the time taken from the client to learn this FML collection. At the end of this step, the success search probability will be computed.

### 6.1 Evolutionary FML Controllers Generation

In this section the algorithm exploited to generate a collection of FML code will be highlighted. As previously mentioned, this algorithm uses theories coming from evolutionary methodologies area, in particular, a modified model of genetic algorithm. More in detail, our approach exploits the classical operators employed in the genetic schema as the crossover and mutation by omitting the selection operator and fitness function concept because they are related to optimization context of genetic algorithm, while, in our approach only the population evolution has to be dealt. In order to start the FML genetic evolution is necessary a collection of ambient intelligence variables on which generate the FML controllers. Let *Variables* be this collection of variables. Each variable, as previously depicted, is characterized by name, universe of discourse, type (input or output) and a collection of fuzzy terms. The pseudocode in listing 5 shows how, starting from *Variables* set, the FML collection is generated:

```
k = number of algorithm iterations
i = 0
while (i < k) {
        FMLCollection = 0
        Choose , randomly , a number n (1 <= n  <= #Variables)
        Choose , randomly , a number n (1 <= m  <= #Variables)
        Extracts n input variables from Variables and put them in Input set .
        Extracts m input variables from Variables and put them in Output set.
        Code the Input variables in FML
        Code the Output input variables in FML
        Choose , randomly , a number r
        Generate r fuzzy rules by , randomly , choosing the fuzzy terms from Input and Output set.
        Add rules to Rules st
        Code the rulebase Rules in FML.
        Add the generated FML program to FMLCollection
        Choose , randomly , a number , g
        j = 0
        while j < g {
                apply g times the genetic operator (crossover and mutation)
                Add the generated FML program to FMLCollection
        }
Upload the FMLCollection on Servers
}
```

**Listing 5.** Evolutionary FML Code Generation

The crossover operator is applied on a pair of rules by crossing the antecedent part of first rule with consequent part of second rule, and vice versa. The mutation operator changes the fuzzy term value of a rule clause (input or output) in a random way. This algorithm generates at most $k \cdot (g+1)$ different FML controllers that, successively, will be coded in RDF and upload on the servers. Reasoning in this way, the client will find the appropriate controllers with probability, at most,

$$\frac{k \cdot (g+1)}{c_1 \cdot c_2 \cdot c_3 \cdot \ldots \cdot c_{(n+m)\cdot r}}$$

where $c_i$ is the number of fuzzy term related to $i^{th}$ fuzzy variable, *n* is the number of clauses in rule antecedent part, *m* is the number of clauses of rule consequent part and r is the total number of rules.

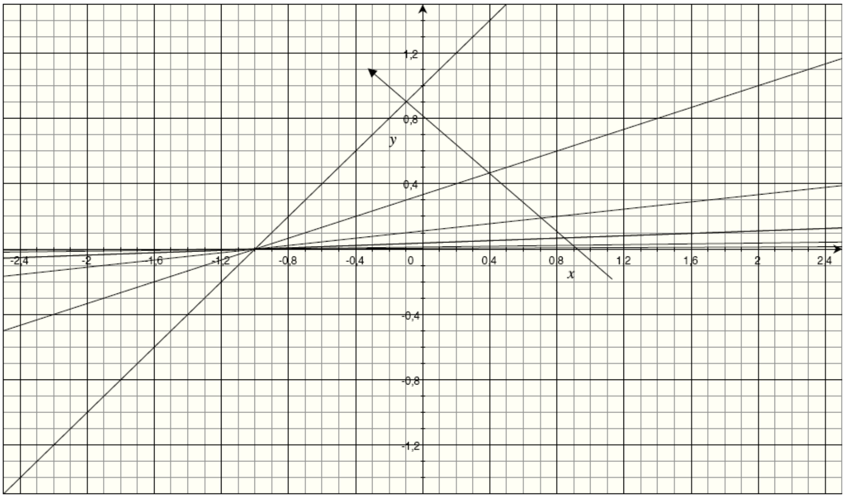


**Fig. 8.** *p = k·(g+1)/325* Probability Map

Our aim is to find the most suitable values of *k* and *g* in order to achieve the following tradeoff: the client finds the appropriate FML controller in the quickest way. In order to evaluate these values is possible to analyze the following simplified mathematical representation of the aforementioned probability:

$$p = \frac{k \cdot (g+1)}{3^{25}}$$

where *n=4*, *m=1*, *r=5* and $c_1 = c_2 = \ldots = c_{25} = 3$. Figure 8 shows the diagram of function *p*, where *g* varies on horizontal axis, *p(g)* varies on vertical axis and *k* represents the map parameter; the arrow individuates the increase direction of *k*. Starting from this graphical analysis it is clear that the are necessary exponential values of the parameter *k* to increase to probability *p* to 1.

In short, the proposed approach could be efficient if the number of server users grows in exponential way regarding the number of queries carried out from client users.

## 7 Related Works and Final Consideration

In recent years, we have witnessed the rapidly growing role of Human-Centric systems as a novel computational paradigm implementing a pervasive framework by focusing on the human and on its interactions with electronic equipments.

Ambient Intelligence can be considered as a new means of distributing network of intelligent devices that provides information, communication, and entertainment around human beings. These systems adapt to the user in a context-aware fashion and differ substantially from contemporary equipment in their appearance in people environments, and in the way users interact with them. The recent developments not only define new market opportunities but also define new challenging tasks for designers, requiring complex AmI systems design as well as strong flexibility and interoperability. This complexity stimulates the developments of sophisticated information technologies skilled to model and realize integrated networks of smart devices where it is possible to dynamically program devices' behavior and making aspects of the programmability accessible to third-party vendors and users. This abstractness is needed to free "control" service, traditionally closed and static inside the device towards more dynamic environments where all devices and services seamlessly interoperate and cooperate with each other.

Several implementations of automatic controllers for AmI environments have been implemented, but only recently the interest of the scientific community in finding appropriate solutions to control large-scale systems has produced an uninterrupted flow of results, some of them involving Fuzzy Logic theories. In [9], a novel type-2 fuzzy system adaptive architecture for agents embedded in ambient intelligent environments (AIEs) is presented. Other approaches have been proposed for the development of learning architectures to devices control in intelligent buildings. In [11] an evolutionary algorithm is analyzed as a candidate for the initial phases of the design of such architectures: fuzzy controllers for the devices are offline induced from data sampled from the environment. Other computational intelligence methodologies have been applied to AmI; for instance, in [18] a novel connectionist embedded agent architecture that combines the use of unobtrusive and relatively simple sensors and employs a constructive algorithm with temporal capabilities which is able to recognize different high level activities (such as sleeping, working at computer, eating) is depicted. Other recent surveys on AmI researches can be found in [7] where an advanced fuzzy-based telecare system is developed. The previous works witness the strategic role played by Fuzzy Logic when applied in a general design methodology applied to complex system. In our approach we follow this trend, renforcing a deeper view abong three actors: the power of fuzzy control (where the power is expressed in terms of user-centered description of control activity), the abstract description level (arising from FML), and the open computational framework that envisage a platform for ubiquitous fuzzy control.

This paper reports our efforts to design and implement a collaborative network system capable of deploying a set of ubiquitous fuzzy granules together with a autonomous composition framework based on semantic web theories offering a method to, dynamically, search and compute the most suitable set of control objects in order to satisfying the user's needs and preferences as required by AmI paradigm. In other words, the proposed framework allows human beings to be considered as the core of a distributed environment capable of adapting itself in order to satisfy the main user's requirements.

## References

1. Acampora, G., Loia, V.: Fuzzy control interoperability for adaptive domotic framework. In: Proceedings of IEEE International Conference on Industrial Informatics, pp. 184–189 (2004)
2. Acampora, G., Loia, V.: Enhancing the fml vision for the design of open ambient intelligence environment. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2578–2583 (2005a)

3. Acampora, G., Loia, V.: Fuzzy control interoperability and scalability for adaptive domotic framework. IEEE Transactions on Industrial Informatics 1(2), 97–111 (2005b)
4. Acampora, G., Loia, V.: Using fml and fuzzy technology in ambient intelligent environments. International Journal of Computational Intelligence Research 1, 171–182 (2005c)
5. Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D.: Web services architecture. In: World-Wide-Web Consortium, W3C (2003)
6. Broekstra, J., Kampman, A., van Harmelen2, F.: Sesame: A generic architecture for storing and querying rdf and rdf schema. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 54–68. Springer, Heidelberg (2002)
7. Clarke, N., Lee, B., Majeed, B., Martin, T.: Long term condition monitoring for tele-care systems. In: Proceedings of IASTED International Conference on Artificial Intelligence and Applications (2005)
8. Gaertner, N., Thirion, B.: A framework for fuzzy knowledge based control. Software: Practice and Experience 30, 1–15 (2000)
9. Hagras, H., Doctor, F., Callaghan, V., Lopez, A.: An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. IEEE Transactions on Fuzzy Systems 15, 41–55 (2007)
10. Lonworks, `http://www.echelon.org`
11. Lopez, A., Sanchez, L., Doctor, F., Hagras, H., Callaghan, V.: An evolutionary algorithm for the off-line data driven generation of fuzzy controllers for intelligent buildings. In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics. Proceedings of IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 42–47 (2004)
12. Mamdani, E.: Applications of fuzzy algorithms for simple dynamic plants applications of fuzzy algorithms for simple dynamic plants applications of fuzzy algorithms for simple dynamic plants. IEE 121, 1585–1588 (1974)
13. Mamdani, E., Assilian, S.: An experience in linguistic synthesis with a fuzzy logic controller. an experience in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine Studies 7, 1–13 (1975)
14. Mozer, M.C.: The Neural Network House: An Environment that Adapts to its Inhabitants. In: Coen, M. (ed.) Proceedings of the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments, pp. 110–114. AAAI Press, Menlo Park (1998)
15. Nijholt, A., Heylen, D.: Multimodal communication in inhabited virtual environments. International Journal of Speech Technology 5, 343–354 (2002)
16. Lassila, O., Swick, R.R.: Resource description framework (rdf) model and syntax specification. W3C recommendation (1999)
17. Piva, S., Marchesotti, L., Bonamico, C., Regazzoni, C.: Context based message selection strategies in a biologically inspired ambient intelligence system. In: Proceedings of Brain Inspired Cognitive Systems (2004)
18. Rivera-Illingworth, F., Callaghan, V., Hagras, H.: A neural network agent based approach to activity detection in ami environments. In: The IEE International Workshop on Intelligent Environments, pp. 92–99 (2005)
19. Rutishauser, U., Joller, J., Douglas, R.: Control and learning of ambience by an intelligent building. IEEE Trans. on Systems, Man and Cybernetics:A, Special Issue on Ambient Intelligence (2004)
20. Seaborne, A.: Rdql - a query language for rdf. W3c member submission, Hewlett Packard (2004)

21. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics 15, 116–132 (1985)
22. Bray, T., Paoli, J., Sperberg-McQueen, C.: W3c recommendation. extensible markup language (xml) 1.0 iii edition. W3C recommendation (1998)
23. Vasilakos, A., Pedrycz, W. (eds.): Ambient Intelligence, Wireless Networking, Ubiquitous Computing. Artech House Press, MA (2006)
24. Wagelaar, D.: Towards a Context-Driven Development Framework for Ambient Intelligence. In: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW 2004), pp. 304–309 (2004)
25. X10, `http://www.x10.org`
26. ZigBee, `http://www.zigbee.org`
27. Zadeh, L.: Fuzzy set. Information Control 8, 338–353 (1965)

# Information Processing in Biomedical Applications

Nick J. Pizzi

National Research Council, Institute for Biodiagnostics, Winnipeg MB, R3B 1Y6, Canada
University of Winnipeg, Applied Computer Science Dept., Winnipeg MB, R3B 2E9, Canada

**Abstract.** To classify biomedical data is to find a mapping from patterns to a set of classes (e.g., disease states). Patterns are represented by features (e.g., metabolite concentrations) and class labels are assigned using a reference test (e.g., an expert's analysis of "normality"). This process often suffers from three significant challenges: voluminous features; pattern paucity; and reference test imprecision. Three computational intelligence based techniques, which exploit the notion of information granulation, are presented to address these challenges. Fuzzy quantile encoding replaces a feature with its membership values in a fuzzy set collection describing the feature's interquantile range. Class label adjustment compensates for reference test imprecision by adjusting design set class labels using a fuzzified similarity measure based on robust measures of class location and dispersion. Stochastic feature selection is a strategy where instances of classifiers are presented with feature regions sampled from an ad hoc cumulative distribution function. These techniques as well as their application to several classification problems in the biomedical domain will be discussed.

**Keywords:** Biomedical Informatics, Biomedical Data Analysis, Information Granules, Pattern Analysis, Computational Intelligence, Feature Selection, Feature Encoding, Fuzzy Set Theory, Biomedical Data Classification, Artificial Neural Networks, Gold Standards Analysis, Performance Measures, Classifier Aggregation, Fuzzy Integration, Fuzzy Systems, Parallel Computing, Information Processing, Biomedical Applications, Granular Computing, Feature Extraction.

## 1 Introduction

Human centric computing has as its main objective the development of computing systems that intuitively adjust to the needs of the user in a seamlessly integrated fashion [27]. This paradigm is relevant to a number of information processing fields including pervasive and ubiquitous computing, ambient intelligence, sensor networks, semantic webs, e-health, e-commerce, wearable hardware, and, specific to our case, biomedical informatics. A typical requirement for human centric computing is a "semantic" layer between fine, detailed numerical data and coarser, generalized abstractions. The semantic layer must perform a translation or transformation from data to abstractions in an efficient and effective manner as the data may need to be abstracted in different ways depending on the needs and objectives of a possibly diverse group of users. In the case of biomedical informatics, for instance, it is necessary to provide effective explanatory analysis while simultaneously finding succinct interpretable (biomedically meaningful) representations. Of course, the challenge is determining the underlying semantic translation that provides the optimal mapping from voluminous data to human manageable, qualitative interpretations. One successful approach to deal with this issue is granular computing.

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 289–311.
springerlink.com

Granular computing [26] is an information processing paradigm that deals with complex information entities in a coherent and comprehensive fashion. Central to this theoretical perspective is the concept of information granules – conceptual entities possessing elements of similarity, functional adjacency, or spatial (or temporal) proximity. Information granules are used to describe or interpret phenomena and carry out processing at the level that is most suitable for the designer of the system and most germane to its potential user. In this sense, one may regard granular computing as an important paradigm for the development of human-centric confirmatory (or exploratory) biomedical data analysis. As a rich theoretical perspective, granular computing subsumes and augments the well established disciplines of interval analysis, fuzzy sets, rough sets, and probability theory [24,25,50,51].

Granular computing research focuses on the construction of a coherent conceptual and methodological framework (and related algorithmic issues): granule quantification and discretization; communication mechanisms between environments of different information granularity levels; translation formalisms between granules grounded in different conceptual environments (e.g., possibility–probability transformations or fuzzy/crisp set approximations); granule construction (e.g., via clustering); and analysis/synthesis of granular systems (e.g., granular classifiers).

There are three types of granulation that are germane to biomedical information processing: discretization; conceptual; and clustering. Discretization involves granulation at the level of feature values. This may be achieved by mapping (binning) a range of values for a biological feature (for example, the concentration of a metabolite) to an ordinal value or through rank ordering of feature values. As it is not feasible to examine the effects of all different discretization combinations for a particular biomedical data analysis problem [18], care must be exercised in designing a heuristic to find near-optimal (or at least adequate) discretizations. Concept granulation involves the notion that different sets of features may give rise to interactions leading to different, possibly, conflicting, higher level conceptual formulations. Clustering involves feature aggregation or transformation to reduce the dimensionality of the original biomedical feature space. Many techniques, with varied relative advantages and disadvantages, fall under this category of granulation: multidimensional scaling, agglomerative techniques, principal component analysis, fuzzy clustering, projection pursuit, independent component analysis, factor analysis, and so on [7,26,49]. For biomedical data analysis, it is important that granulation techniques do not mask or diminish the information content present in the original (biomedically relevant) feature space.

### 1.1 Biomedicine and Vagueness

Imprecision, incompleteness, and uncertainty are intrinsic to the practice of medicine. While this art of making decisions with inadequate information is often impervious to precise modes of analytical reasoning, it is regularly amenable to approximate ones [15], and, as a result, the field has become a fertile and active domain with which to exercise granular computing based modes of reasoning. Further, medical decision-making is paradigmatic of general decision support systems in which principles, procedures, data, and knowledge are approximate; hence, successful methods applied to the medical domain may often be generalized across many application domains.

The medical diagnostic process involves an inference of a disease from a set of symptoms based on a body of medical knowledge about nosology[1] and symptomatology[2]. Unfortunately, vagueness is a hallmark of this process. A disease may manifest itself differently from one patient to the next as well as temporally for the same patient. Medical diagnosis is confounded by multiple diseases present in a particular patient or a specific symptom present in multiple disease states. A patient's historical information may be incomplete, physical examinations may inadvertently ignore symptoms, laboratory test results may be imprecise, and the distinction between the states of normality and abnormality is not necessarily crisp. Important diagnostic information acquired from medical instrumentation such as magnetic resonance, infrared, or mass spectrometers is often complex and voluminous and their interpretation may vary from one expert to the next. Medical knowledge is often couched in necessarily imprecise linguistic terminology. The proliferation of new medical knowledge, introduces uncertainty and inconsistencies during its assimilation into the current orthodoxy. For instance, the inclusion of new diagnostic procedures after they have been successfully assessed against the corresponding external reference test (this currently accepted diagnostic procedure is often referred to as the "gold standard"), which itself may be imprecise.

## 1.2 Granular Computing and Computational Intelligence Strategy

Classifying biomedical data involves finding a mapping (relationship) from patterns (e.g., data relating to some type of tissue or biofluid) to a set of classes (e.g., disease states). Patterns are represented by features (e.g., concentrations of biological compounds) and class labels are assigned using a reference test (e.g., a medical expert's analysis of tissue being "normal" or "abnormal"). This process often suffers from three significant challenges: the number of features in a pattern is high; the number of patterns is low; and the reference test may be unreliable. The first two challenges, known collectively as the "curse of dimensionality", cause an inability to find robust, general solutions. This is often addressed by reducing, in some fashion, the number of features; however, a direct correspondence back to the original features is necessary for medical experts to make informed judgments about the mapping's predictive power. While external reference tests may be well-established benchmarks, they are seldom perfectly accurate and sometimes improperly applied. Nevertheless, any strategy that compensates for this reference test imprecision must ensure that the mapping is correctly validated against the benchmark. In this chapter we present three techniques based upon computational intelligence and the paradigm of granular computing to deal with the biomedical data analysis challenges described above.

*Fuzzy quantile encoding* is a classification preprocessing method that replaces a specific feature value for a pattern with its membership values in a collection of fuzzy sets describing the interquantile range of all values for that feature within the dataset. This method "normalizes" features to the unit interval, diminishes the impact of feature "outliers", and improves the overall accuracy and computational performance of adaptive classifiers such as supervised feed-forward neural networks.

[1] The branch of medical science dealing with the classification of diseases.

[2] The study of symptoms of disease and signs of pathogens for the purpose of diagnosis.

*Gold standard class label adjustment* is a set of mitigation strategies that compensates for possible reference test imprecision by adjusting the design set class labels using a fuzzified similarity measure based on robust measures of location and dispersion of class medoids (robust centroids). These mitigation strategies fall into three categories: reassignment involves changing the class label of a design subset pattern, if it is found to be more “similar” to patterns from another class; surrogation involves using a new space of class labels for the design set (for instance, cluster analysis may indicate that patterns in a particular class are distributed in such a way that they are better represented by two surrogate class labels); and gradation involves the fuzzy set notion of a pattern belonging to all classes to varying degrees (that is, moving from a crisp, Boolean class assignment to a fuzzy one).

*Stochastic feature selection* is a parallelized classification strategy where many instances of heterogeneous classifiers are presented with (possibly quadratically transformed) feature regions of varying cardinality. Regions are stochastically sampled from an ad hoc cumulative distribution function that is iteratively updated based on a frequency histogram of features used by prior classifiers whose performance (accuracy) exceeds a pre-defined threshold. Fuzzy integration is used to aggregate the best classification outcomes.

The schema presented in Figure 1 indicates the three main classification phases: pre-processing, fuzzy quantile encoding and gold standard class label adjustment;

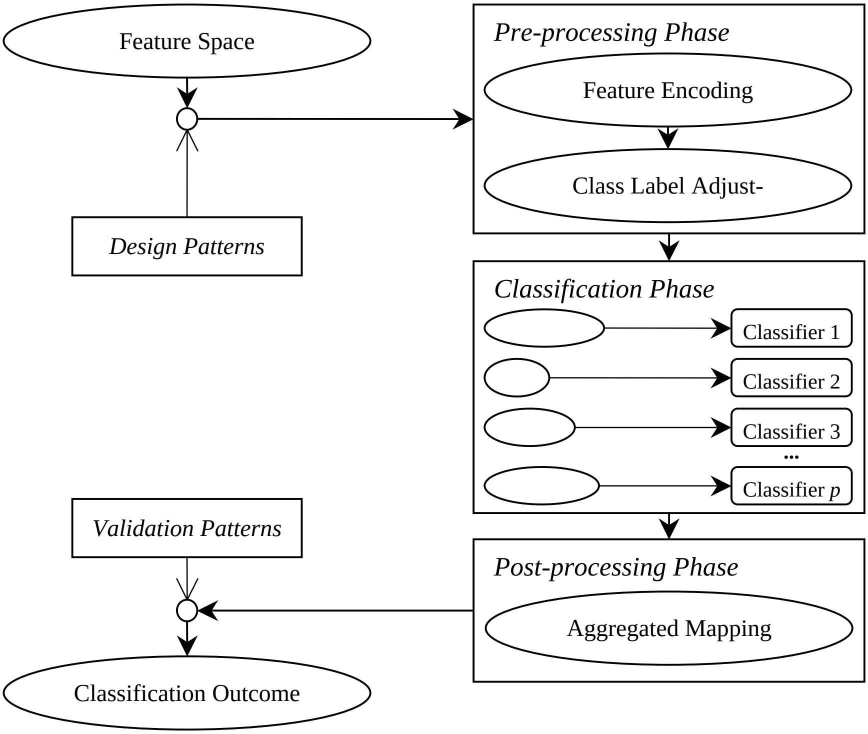


**Fig. 1.** Schema indicating the pre-/post-processing and classification phases

classification, stochastic feature selection coupled with a set of $p$ heterogeneous classifiers; and, post-processing, mapping aggregation (prediction fusion). The figure also clearly indicates that only design patterns (those patterns randomly assigned to the design subset) are used in these phases to construct the aggregated classification mapping. In order to attenuate bias, classification performance (accuracy) is assessed using this mapping with only the validation patterns.

Each of these classification techniques and strategies will be discussed in the sections that follow. A description of the architecture of the biomedical data analysis software, which implements several key aspects of this methodology, will be provided. Finally, the successful application of this methodology to several classification problems in the biomedical domain will be discussed.

## 2 Biomedical Data Classification

The latest biomedical spectroscopic modalities produce information rich but complex and voluminous data [23]. For instance, magnetic resonance (MR) spectroscopy, which exploits the interaction between an external homogenous magnetic field and a nucleus that possesses spin, is a reliable and versatile spectroscopic modality [10]. Coupled with robust multivariate discrimination methods, it is especially useful in the classification and interpretation of high-dimensional biomedical spectra of biofluids and tissues [42]. However, the sample to feature ratio of these data is typically low; the feature space dimensionality is $O(10^3–10^4)$ while the sample size is O(10–100). This “curse of dimensionality” is a serious challenge for the classification of biomedical spectra: the excess degrees of freedom tend to cause overfitting, which affects the reliability of the chosen classifier.

Advances in pattern recognition, computational intelligence, and granular computing, contribute ever more sophisticated models upon which to build ever more sophisticated classifiers. Herein lies a major problem: if these models are highly non-linear, they may be unstable, if they are iterative, they may not converge, if they are probabilistic, they may be based on underlying statistical assumptions that are often not true in real-world scenarios. Preprocessing may address these concerns: data may be transformed such that a non-linear model may be replaced by a linear one, the dimensionality of the data may be reduced so that an iterative method may converge or may be substituted for an analytic one, or the data may be “normalized”, in some sense, such that the underlying statistical assumptions of a probabilistic model are realized. Years of investigations in biomedical data analysis have led to this author’s conjecture that the 80/20 rule holds in the development of classification systems: 20% of a researcher’s effort should be spent on selecting and tuning a classifier; 80% should be spent on a thorough analysis of the data to simplify them, via pre-processing, prior to presentation to the classifier of choice.

Here, we formally introduce some notation used throughout this chapter. Let $\boldsymbol{X}=\{(\boldsymbol{x}_k,\omega_k), k=1..N\}$ be a set of $N$ patterns, $\boldsymbol{x}_k \in \Re^n$, with respective class labels, $\omega_k \in \Omega$, $\Omega=\{1..c\}$ that are randomly assigned to either a design subset, $\boldsymbol{X}^D$, comprising $N_D$ patterns, or a validation subset, $\boldsymbol{X}^V$, comprising $N_V$ patterns $(N_D+N_V=N)$. Classification involves finding a mapping (function approximation), $f:\boldsymbol{X}^D \rightarrow \Omega$, and then validating its effectiveness using $\boldsymbol{X}^V$, $f:\boldsymbol{X}^V \rightarrow \Omega$ (if the predicted class label does not match the assigned class label then it is considered to be a misclassification).

Many classification architectures exist (granular classifiers, supervised artificial neural networks, multivariate statistical methods, evolutionary computation, hybrid strategies, and so on), with various advantages and disadvantages [7]. However, as mentioned above significant effort must be expended in the analysis of appropriate pre-processing strategies. For instance, feature selection is a typical preprocessing strategy for attenuating the effects of the curse of dimensionality by reducing the size of the input (feature) space. Feature selection involves finding a mapping, $f$:X→X′, where $X' \subseteq \Re^m$ ($m << n$) is the reduced feature space. Subsequently, classification involves finding a mapping from the reduced feature space to the space of class labels, $g$: X′→Ω. The intent of this strategy is to select those features possessing significant discriminatory power.

One aspect of biomedical data classification that is often glossed over is the reliable validation of the accuracy results generated by a classification schema. It is essential that datasets be divided (randomly) into design and validation sets. Design patterns may be used in the construction of a classification system but once this phase is complete, performance results must be based on the validation patterns. Given this necessary condition, how is the performance of a classification system to be measured given a $c \times c$ confusion matrix of desired versus actual class labels using the validation patterns? The conventional performance measure is the ratio of correctly classified patterns to the total number of patterns, $P_O$

$$P_o = N_V^{-1} \sum_i n_{ii} \; (i = 1, \ldots, c) \tag{1}$$

where $n_{ij}$ is the number of class $i$ validation patterns predicted to belong to class $j$. An alternate performance measure is the average class-wise accuracy, $P_A$

$$P_A = c^{-1} \sum_i \left( n_{ii} \middle/ \sum_j n_{ij} \right) (i, j = 1, \ldots, c) \tag{2}$$

But neither $P_o$ nor $P_A$ take into account any agreement due to chance [8], $P_L$

$$P_L = N^{-2} \sum_i \left( \sum_j n_{ij} \sum_j n_{ji} \right) (i, j = 1, \ldots, c) \tag{3}$$

A more conservative performance measure is the κ score [9], a chance-corrected measure of agreement between the desired and actual class assignments

$$\kappa = \left( P_o - P_L \right) \big/ \left( 1 - P_L \right) \tag{4}$$

If the agreement is due strictly to chance, κ=0. If it is greater than chance κ>0; κ=1 indicates complete agreement. If the agreement is less than chance then κ<0 (floor depends upon the marginal distributions). A useful benchmark for agreement strength (confidence) is: poor (κ=0), slight (0.0<κ≤0.2), fair (0.2<κ≤0.4), moderate (0.4<κ≤0.6), substantial (0.6<κ≤0.8), and almost perfect (0.8<κ<1.0) [17]. Table 1 shows the necessity of careful analysis of accuracy. It lists two confusion matrices with the same number of patterns per class and the same overall accuracy, $P_O$=0.66. Using $P_A$ it is clearer that the accuracy is in fact worse in Table 1(ii), $P_A$=0.33 than Table 1(i) $P_A$=0.62. However, via κ, it is clear that the apparent accuracy in the second

**Table 1.** Two three-class confusion matrices with the same pattern distributions

| | (i)C1 | C2 | C3 | (ii)C1 | C2 | C3 | N=300 |
|---|---|---|---|---|---|---|---|
| C1 | 15 | 10 | 5 | 3 | 24 | 3 | $N_1$=30 |
| C2 | 37 | 163 | 40 | 24 | 192 | 24 | $N_2$=240 |
| C3 | 2 | 8 | 20 | 3 | 24 | 3 | $N_3$=30 |
| Accuracy | $P_O$=0.66 | $P_A$=0.62 | κ=0.29 | $P_O$=0.66 | $P_A$=0.33 | κ=0.00 | |

confusion matrix is due strictly to chance, κ=0.00 versus κ=0.29. This is further evidenced by examining $P_L$ (0.52 versus 0.66).

Here, we briefly present three classifiers (two neural networks and one statistical method) that have been used for biomedical data analysis described in the applications section. Neural networks [1] are self-adaptive, machine learning systems composed of layers of processing elements, which are sets of inputs and weights combined to generate outputs used by an adjacent layer. Supervised networks [38] require the desired class labels for each pattern so that they may be compared to the predicted label. Based on these comparisons, a learning strategy, used to make incremental changes to the weights, minimizes an error criterion.

The multi-layer perceptron (MLP) [39] is a supervised feed-forward network, which has consistently demonstrated its effectiveness as a reliable nonlinear classification technique [3]. The transfer function $\gamma$ (often the logistic function, $\gamma(x)=(1+e^{-x})^{-1}$) is sigmoidal and the output of processing element $j$ is $x_j=\gamma(\sum_i w_{ji}x_i)$. In general, an MLP may be considered a non-linear regression system that performs a gradient descent search through the weight space, searching for minima.

The probabilistic neural network (PNN) [43] uses patterns to construct probability density functions (pdf) to estimate the likelihood of a given pattern belonging to a class. When the class pdfs are known, a PNN correspond to a Bayesian classifier. Since true class pdfs are rarely known, they are usually approximated via a sampling histogram and Parzen estimators [22]. This involves the construction of unit area Gaussians centred at the values of the features for every design pattern. These Gaussians are summed and scaled to produce a composite curve. As the number of design patterns increase, the composite curve asymptotically approaches the true pdf. [However, it is not possible to determine the number of patterns required to estimate the pdf to a specified accuracy.]

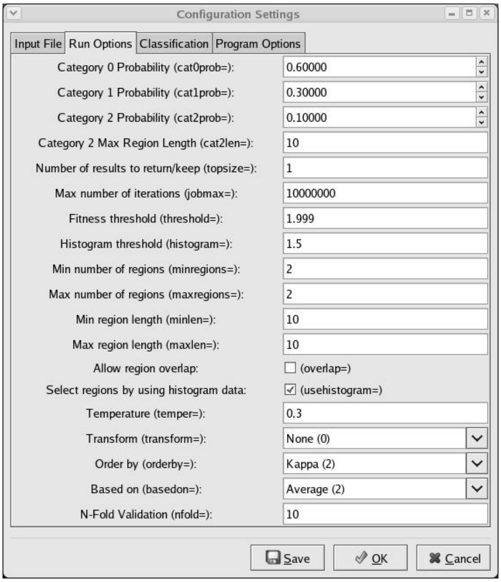
Linear discriminant analysis (LDA) is a classification approach that determines linear decision boundaries between $c$ classes while taking into account inter- and intra-class variances [40]. If the error distributions for each class are the same, LDA constructs the optimal linear decision boundary between the classes. In real-world situations, this optimality is seldom achieved since different classes typically give rise to different distributions. LDA is a useful linear classifier; however, when appropriate data preprocessing is applied, in particular, dimensionality reduction techniques such as stochastic feature selection. LDA allocates a pattern, $\boldsymbol{x}$, to class $i$ for which the probability distribution, $p_i(\boldsymbol{x})$, is greatest. That is, $\boldsymbol{x}$ is allocated to class $i$, if $q_ip_i(\boldsymbol{x})>q_jp_j(\boldsymbol{x})$ ($\forall j\neq i$), where $q$ are the prior (or proportional) probabilities. The discriminant function is $L_i(\boldsymbol{x})=\log q_i+\boldsymbol{m}_i^{\mathrm{T}}W^{-1}(\boldsymbol{x}-\tfrac{1}{2}\boldsymbol{m}_i)$ where $\boldsymbol{m}_i$ is the mean for class $i$ and $\boldsymbol{W}$ is the covariance matrix.

## 3 Stochastic Feature Selection

Stochastic feature selection (SFS) is a feature selection/reduction pre-processing method that is tightly coupled to the classification phase. SFS may be used with any homogeneous or heterogeneous set of classifiers (e.g., LDA, MLP, PNN, or support vector machines [47]). Essentially, SFS iteratively presents, in a highly parallelized fashion, many feature regions (contiguous subsets of pattern features) to the set of classifiers retaining the best set of classifier/region pairs. Figure 2 lists several of the key parameters used in SFS, which we will reference in the following detailed description of SFS.

After selecting the minimum and maximum number of feature regions and the minimum and maximum size (cardinality) for a feature region (cf. fields shown in Figure 2, “Min number of regions”, “Max number of regions”, “Min region length”, “Max region length”, respectively), the general procedure is: (i) randomly select a number of feature regions and, for each region, select a random size (satisfying the above constraints); (ii) prune the features not selected in (i) from the training and monitoring sets; (iii) use the training set and classifier to produce classification coefficients; (iv) test these candidate coefficients with the monitoring set; (v) repeat steps (i)–(iv) until either the accuracy threshold (“Fitness threshold”) or maximum number of iterations (“Max number of iterations”) is exceeded; (vi) finally, use the best coefficients found and assess their performance using the validation set. Note that the training and monitoring sets are composed of patterns exclusively from the design set. The validation patterns are only used in step (vi).

SFS retains a list of the best classification results (“Number of results to return/keep”) based on the selected fitness function (“Order by”). The fitness function



Category 0 Probability (cat0prob=): 0.60000
Category 1 Probability (cat1prob=): 0.30000
Category 2 Probability (cat2prob=): 0.10000
Category 2 Max Region Length (cat2len=): 10
Number of results to return/keep (topsize=): 1
Max number of iterations (jobmax=): 10000000
Fitness threshold (threshold=): 1.999
Histogram threshold (histogram=): 1.5
Min number of regions (minregions=): 2
Max number of regions (maxregions=): 2
Min region length (minlen=): 10
Max region length (maxlen=): 10
Allow region overlap: □ (overlap=)
Select regions by using histogram data: ☑ (usehistogram=)
Temperature (temper=): 0.3
Transform (transform=): None (0)
Order by (orderby=): Kappa (2)
Based on (basedon=): Average (2)
N-Fold Validation (nfold=): 10

**Fig. 2.** Several parameters used for stochastic feature selection

may be $P_O$, $P_A$, or κ, which may be applied ("Based on") exclusively to the training set or in conjunction with the monitoring set and internal cross-validation is also used ("N-Fold Validation"). Feature regions are normally disjoint but this can be relaxed ("Allow region overlap"). Moreover, transformations may be performed on regions ("Transform") such as computing their average feature value, their variance, or other statistical moments.

### 3.1 Feature Frequency Histogram

The stochastic nature of this method is normally controlled by the feature frequency histogram (see Figure 3). During an SFS run, the performance of each classification task is assessed using the selected fitness function. If the fitness exceeds the histogram fitness threshold (cf. Figure 2, "Histogram threshold"), which is set to some value less than the fitness threshold stopping criterion, the frequency histogram is incremented at those feature indices corresponding to the regions used by the particular classification task. This histogram is then used to generate a cumulative distribution function (cdf). Now, when feature regions are selected for a new classification task, features are randomly selected using the current cdf. So, rather than each feature having an equal likelihood of being selected for a new classification task, those features that were used in previous "successful" tasks have a greater likelihood of being chosen. A temperature term, $t \in [0,1]$, provides additional control over this process. If t=0, the cdf is used as described but, as t→1, the randomness becomes more uniform (when t=1 a strict uniform distribution is used). A useful interactive option is to pause SFS, select those regions that have been shown to be most discriminatory, and continue SFS so that subsequent regions will be selected only from these highly discriminatory features.

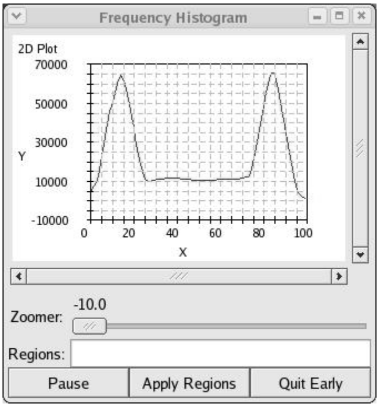


**Fig. 3.** A typical SFS feature frequency histogram

### 3.2 Quadratic Combination of Features

A useful SFS pre-processing option is to augment the original features with a quadratic combination of feature regions. The intention here is that if the original feature space possesses non-linear decision boundaries between classes, the new (quadratic) parameter space may possess more “linearized” decision boundaries. For instance, say we have a set of three-feature two-class points (patterns), $\boldsymbol{x}=\{x_1,x_2,x_3\}\in[0,1]^3)$ bounded by the unit hypercube where one class of points, $\omega_1$, are those within the unit hypersphere $(x_1^2+x_2^2+x_3^2<1)$ and the other class, $\omega_2$, are those points outside $(x_1^2+x_2^2+x_3^2\geq 1)$. These patterns are obviously separated by a circular (non-linear) decision boundary. A linear classification system using, for instance, linear discriminant analysis, would perform poorly ($P_O\approx 0.50$) with such a dataset as no linear decision boundary (plane) can accurately delineate the two classes of points (patterns). However, if we create a new three-coordinate feature space by simply squaring the original features, the decision boundary (in this new space) would be a plane and a linear classifier will now perfectly separate the two classes of patterns.

SFS has three categories of quadratic combinations with which to augment the original features (cf. the respective fields shown in Figure 2): (i) using the original feature region (“Category 0 Probability”); (ii) squaring the values for the selected feature region (“Category 1 Probability”); or (iii) using all pair-wise cross-products of features from two regions (“Category 2 Probability”). Given the potential combinatorial explosion with the third category, an upper limit for the region size may also be specified (“Category 2 Max Region Length”). The probabilities of selecting one of these quadratic combination categories must sum to 1.0.

### 3.3 Parallelized Classification

SFS takes full advantage of parallel computations using the Scopira Agent Library [6], a sophisticated message-passing library similar in functionality to MPI [41]. Given a high-performance computing cluster (e.g., a Linux Beowulf cluster) environment, classification tasks are distributed to slave nodes for computation. A master node coordinates the distribution of tasks, updates the feature frequency histogram, and records intermediate classification performance results. To minimize interprocess communication and maximize continuous computational loads on the processors, SFS efficiently “bundles” sets of classification tasks. Furthermore, while SFS exploits parallelism, it still remains a strictly deterministic system. That is, experimental results are perfectly reproducible regardless of computational load, which is extremely important in the analysis, and interpretation of complex biomedical data.

## 4 Fuzzy Quantile Encoding

Zadeh’s seminal work on fuzzy set theory [51] may be applied to a classification preprocessing technique that encodes the feature space prior to presentation to a classifier. For instance, a feature may be intervalized across a collection of fuzzy sets thereby producing a list of degrees of membership for each of the fuzzy sets. In other words, given $s$ fuzzy sets, $F_1$, $F_2$, …, $F_s$, and $f_i$ is the membership function for fuzzy set $i$, then the list of values for a single feature value $x$ is $\{f_1(x), f_2(x), \ldots, f_s(x)\}$. Figure 4 illustrates this intervalization approach using the membership functions for two fuzzy sets for feature $i$ that overlap at 0.5 (see below).

Fuzzy quantile encoding (FQE) uses a feature's quantile values as the consecutive intersections of triangular (or trapezoidal) fuzzy sets [36]. To derive the formula (a full derivation and complete discussion may be found in [30]), the following terms need to be defined. Let *b*, $0 \le b \le 1$, be the boundary value at the intersection of the fuzzy sets. For simplicity, *b* may be held constant for each intersection. Let *w* be the width of the top of the trapezoid of the fuzzy sets. [If $w=0$, the $f_i$'s are triangular fuzzy sets.] Let $l_i$ and $r_i$ be the left and right boundary, respectively, of the fuzzy set $F_i$ such that $f_i(l_i)=f_i(r_i)=b$. Finally, let *x* be the original non-encoded input value. Then,

$$f_i(x) = \begin{cases} 1 \wedge \left( 0 \vee \left( 1 + w - 2\dfrac{1+w-b}{r_i - l_i} \left| x - \dfrac{l_i + r_i}{2} \right| \right) \right) & l_i < r_i \\ 1 & l_i = r_i = x \\ 0 & l_i = r_i \ne x \end{cases} \tag{5}$$

where $\vee$ and $\wedge$ are the max and min operators, respectively (other norm/co-norm pairs, of course, are permissible). The latter two cases define a delta function when $l_i=r_i$. These delta functions satisfy the criteria for a fuzzy set: it is monotonic and it maps onto the unit interval. Delta functions may arise when pattern feature values are significantly skewed (non-normal). It is important to note that, since $f_i(r_i)=f_{i+1}(l_{i+1})=b$, $r_i=l_{i+1}$ ($\forall i=1..s-1$). It should also be noted that the corresponding membership functions are symmetric about the boundaries $l_i$ and $r_i$. When $b \ge 0.5$ and $w=0$ there exists a strict 1–1 correspondence between the encoding and the original feature value. When $b<0.5$ (or $w>0$), a 1–many correspondence exists.

Quantiles are used to determine reasonable values for the fuzzy set boundaries $l_i$ and $r_i$. The $Q^{\text{th}}$ quantile of *N* feature values is a value such that *Q*% of the area under the relative frequency distribution for the feature values lies to the left of the $Q^{\text{th}}$ quantile and (100–*Q*)% of the area under the distribution lies to its right.

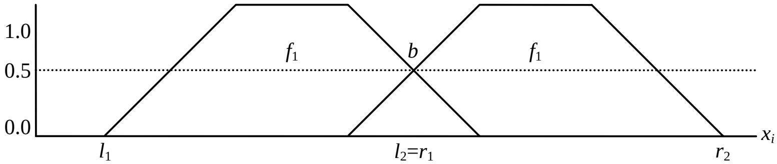


**Fig. 4.** FQE membership functions using two fuzzy sets for feature *i*

## 4.1 Interquartile Range

Normally, the selected quantiles for FQE are the feature's quartiles (see Figure 5 illustration): the lower quartile (25th quantile), $Q_L$; the median (50th quantile), *m*; and the upper quartile (75th quantile), $Q_U$. By using the interquartile range for the feature *j*,

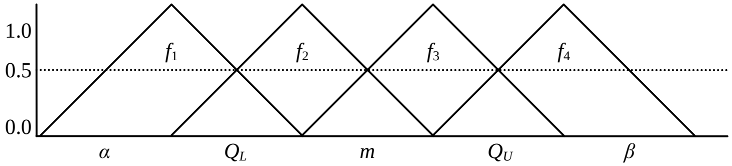


**Fig. 5.** FQE membership functions using a feature's interquartile range

uniform coverage is effected through the use of four overlapping fuzzy sets, $F^j{}_1$, $F^j{}_2$, $F^j{}_3$, $F^j{}_4$. To ensure a 1–1 mapping between the original feature values and the FQE values, $w$=0 and $b$=0. [However, the constraint on $w$ can be relaxed (see sub-section below).] Specifically, the membership functions for feature $j$ are

$$\begin{aligned} f_1^j(x_j) &= 1 \wedge \left[0 \vee \left[1 - \left| x - 0.5\left(\alpha^j + Q_L^j\right)\right| / \left(Q_L^j - \alpha^j\right)\right]\right] \\ f_2^j(x_j) &= 1 \wedge \left[0 \vee \left[1 - \left| x - 0.5\left(Q_L^j + m^j\right)\right| / \left(m - Q_L^j\right)\right]\right] \\ f_3^j(x_j) &= 1 \wedge \left[0 \vee \left[1 - \left| x - 0.5\left(m^j + Q_U^j\right)\right| / \left(Q_U^j - m^j\right)\right]\right] \\ f_4^j(x_j) &= 1 \wedge \left[0 \vee \left[1 - \left| x - 0.5\left(Q_U^j + \beta^j\right)\right| / \left(\beta^j - Q_U^j\right)\right]\right] \end{aligned} \tag{6}$$

where $\alpha^j$ and $\beta^j$ are the feature's respective minimum and maximum values. A dimension-preserving variant to this fuzzy encoding approach [29] is to use a single membership function, $f_j(x)$, which corresponds to a piece-wise linear fuzzy set ($w$=0), to capture the information represented by the feature's interquartile range

$$f^j(x) = \begin{cases} b(x-\alpha)(Q_L-\alpha)^{-1} & \text{if} \quad \alpha \le x < Q_L \\ (1-b)(x-Q_L)(m-Q_L)^{-1} + b & \text{if} \quad Q_L \le x < m \\ (b-1)(x-m)(Q_U-m)^{-1} + 1 & \text{if} \quad m \le x < Q_U \\ -b(x-Q_U)(\beta-Q_U)^{-1} + b & \text{if} \quad Q_U \le x < \beta \\ 0 & \text{if} \quad x < \alpha \vee x > \beta \end{cases} \tag{7}$$

### 4.2 Dispersion Adjustment

An effective extension to fuzzy quantile encoding involves adjusting the fuzzy sets in order to take into account a feature's overall dispersion of values [28]. A robust technique to implement dispersion-adjusted FQE is to use a feature's median of absolute deviations, τ

$$\tau(x) = \frac{m(|x - m(x)|)}{\sigma} \tag{8}$$

where $m$ is the feature's median and $\sigma$=0.6745 to ensure that, as the error distribution becomes more normal, τ converges to the standard deviation. While only 40% efficient for normal data [13], τ is robust to outliers and long-tailed distributions. In other words, as the features becomes more contaminated (less normal), the relative efficiency of τ becomes greater than the standard deviation.

In order to take into account a feature's overall dispersion, the constraint on $w$ needs to be relaxed; for a given pattern feature, let $w$=τ. Using (8), (5) can easily be modified to now permit the use of trapezoidal fuzzy sets. As the dispersion increases (τ becomes larger), the width of the trapezoid increases and, as a result, more original feature values will be encoded to 1. As the dispersion decreases, the trapezoid approaches a triangular fuzzy set, so fewer values will be encoded to 1.

### 4.3 FQE Properties

FQE may be easily integrated into any classification system. The input layer (feature space) will have $s$×$n$ coordinates where $n$ is the dimensionality of the original feature space and $s$ is the number of fuzzy sets used for encoding ($s$=4 for interquartile encoding). FQE exhibits several useful properties.

First, the feature space is "normalized": that is, for any given pattern feature, $x$, its corresponding membership functions map feature values onto the unit interval, $f_i(x)\in[0,1]$ ($\forall i$=1..$s$). This is particularly useful during the classification process since scaled biomedical data stabilize the effects of extreme variance disparities across pattern features [38]. Without scaled data, features with large variances will have a tendency to predominate, during the training phase, over those features with small variances even though the latter features may be highly discriminatory.

Another beneficial property is that, during the construction of the discriminating class decision boundaries, feature values that may be considered as outliers impact less severely upon classifiers that employ any type of iterative adjustments to its error function (e.g., artificial neural networks such as MLP). This does not mean that patterns with features that are outliers are removed during the design or validation phases of the classification process, however. FQE values will approach zero as values move outside a feature's interquartile range. In the case of MLP, where its hidden layer processing elements are summing products of weights and input values this is important since, if the FQE values of an outlier are all zero or near zero, those values will contribute very little to the learning process (local error adjustments) regardless of the processing elements weights. This is an extremely useful property if the original feature value is indeed an outlier (nevertheless, if it is not an outlier it still does contribute to a degree). Conversely, values that are within the feature's interquartile range will contribute strongly to the iterative learning process.

Another purpose behind FQE intervalization, as with any type of intervalization, is to reduce the effects of noise in the data as well as to transform the problem in such a way that a non-linear regression model such as MLP can provide better (more accurate) solutions.

Moreover, a FQE based classifier projects the original $n$-dimensional pattern feature space onto a new $4n$-dimensional parameter space of membership values. This projection often has the positive effect of "linearizing", to some degree, the discrimination problem (that is, moving from non-linear to linear class decision boundaries).

Further, since many FQE values are zero (or near zero), artificial neural network processing elements that use these encoded values as input terms will produce output values that are also at or near zero regardless of the corresponding processing element weights. Subsequently, these processing elements tend to contribute little to the overall classification error (and, derivatively, to the overall learning) of the FQE-based artificial neural network so resultant errors propagated back through the neural network are not caused (to any great extent) by these values. These simplifications, caused by the projection, often significantly reduce the training phase convergence time for supervised artificial neural networks [29].

## 5 Fuzzy Class Label Adjustment

Gold standard fuzzy class label adjustment (GSA) compensates for the possible imprecision of a well-established but tarnished gold standard (external reference test) by adjusting, if necessary, the class labels of the design set patterns. The procedure begins with finding the centroids of each class using their respective design set patterns. Distances are computed between each design pattern and each class centroid. A fuzzy set theoretic membership function uses these distances to adjust the class labels; in general, the further a pattern is from a class centroid, the lower its membership value for that class. However, the class label for a pattern will only be adjusted if it is sufficiently distant from the centroid of its original class and sufficiently near another class' centroid. Note that any adjustments made to the gold standard occur only for patterns in the design set; for verification purposes, the class labels for the validation set patterns are never altered. Hence, the efficacy of this method is always measured against the original gold standard (regardless of its possible imprecision).

Distances and dispersions are measured using robust multivariate statistics since they are much more resistant to effects caused by extreme feature values than parametric statistics. More specifically, a statistical estimate is robust if it is insensitive to slight deviations from its requisite model assumptions (often normal assumptions) about the underlying feature distribution [13]. This is crucial when dealing with outliers, patterns that do not follow the distribution of the majority of the data.

Although it is a univariate estimator, $\tau(x)$ (see (8)) may be extended to the multivariate case by computing a vector, $\tau^l=[\tau^l_1..\tau^l_n]$, which is a feature-wise measure of dispersion for the class $l$ patterns. First, let $\boldsymbol{X}^l=\{(\boldsymbol{x}_k,l),\ k=1..N_l\}\subset\boldsymbol{X}$ be the set of all patterns belonging to class $l$ ($N_l$ is the number of class $l$ patterns and $\boldsymbol{x}_k=[x_{k1}..x_{kn}]$). Also, let $\boldsymbol{z}^l_j=[x_{ij}]$ ($i=1..N_l$) be the respective values of feature $j$ for the class $l$ patterns. Now, $\tau^l_j=m|\boldsymbol{z}^l_j-m(\boldsymbol{z}^l_j)|/0.6745$. The distance between each pattern and each of the class centroids may now be determined. The weighted distance, $d^l$, of $\boldsymbol{x}_k$ from the class $l$ centroid (more correctly its medoid) may be defined as

$$d^l(\boldsymbol{x}_k)=\sum\nolimits_j\left|\frac{x_{kj}-m(z^l_j)}{\tau^l_j}\right| \tag{9}$$

This distance measure is incorporated into the original gold standard using membership functions; the class $l$ membership function for a pattern, $\boldsymbol{x}_k$, is defined as

$$f_l\left(x_k\right)=\left[1+\left(d^l(\boldsymbol{x}_k)/q\right)^p\right]^{-1} \tag{10}$$

where $p$>1 and $q$>0 describe the shape and amount of fuzziness for the membership function ($0 \le f_l(\boldsymbol{x}_k) \le 1$). Figure 6(i) plots (10) for different values of $p$ with a constant $q$. Note that $f$ is sigmoidal and that as $p$ increases, $f$ approaches a step function. The point, at which the membership function is 0.5, occurs when the distance equals $q$. Figure 6(ii) plots $f$ for different values of $q$ with a constant $p$. As $q$ increases, membership values will remain high even at great distances.

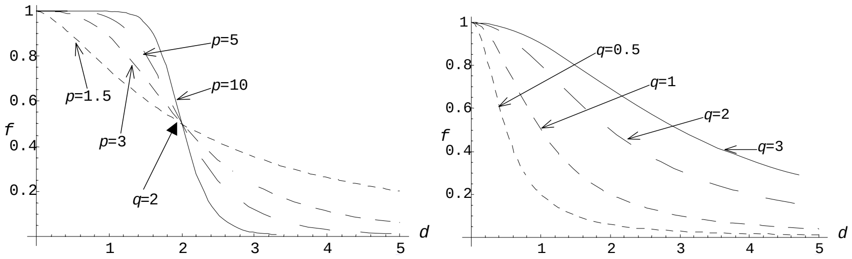


**Fig. 6.** Plot of $f$ versus distance ($d$) between a pattern and a class medoid with (i) varying $p$ ($q$=2) and (ii) varying $q$ ($p$=2)

Finally, we use contrast intensification, $y_l$, on the class $l$ membership function to increase membership values above 0.5 and reduce those values that are below this point [51]. Using GSA, we may now recode the class label for $\boldsymbol{x}_k$ from the scalar $\omega_k$ to the vector [$y_i$] ($i$=1..$c$).

$$y_l(\boldsymbol{x}_k)=\begin{cases} 2f_l^2\left(\boldsymbol{x}_k\right) & if\ \ 0\le f_l\left(\boldsymbol{x}_k\right)\le 0.5 \\ 1-2\left(1-f_l\left(\boldsymbol{x}_k\right)\right)^2 & if\ 0.5\le f_l\left(\boldsymbol{x}_k\right)\le 1.0 \end{cases} \tag{11}$$

If $\boldsymbol{x}_k$ was originally assigned to, say class $l$, by the gold standard, it may be the case that it was, in fact, closest to some other class medoid, say class $o$. In this case, $y_o(\boldsymbol{x}_k)>y_l(\boldsymbol{x}_k)$ and, hence, the original gold standard assignment will no longer predominate. If this is undesirable (or unacceptable) for the particular problem domain, the situation may be rectified by constraining the membership functions expressed by (11) so that $f_l(\boldsymbol{x}_k)=f_o(\boldsymbol{x}_k)+\varepsilon$ where ε is a small positive constant. Now, a pattern will never be reassigned to a class different from the one to which it was originally assigned. However, if a pattern is sufficiently near another class medoid then the corresponding class membership value for that pattern will not be zero. In general, the further $\boldsymbol{x}_k$ is from a class medoid, the lower its membership value for that class. While the original class label assigned by the gold standard is crisp ($\boldsymbol{x}_k$ belongs to one and only one class with degree 1), the (soft) class label assigned by GSA (using (11)) is fuzzy ($\boldsymbol{x}$ belongs to all classes to varying degrees).

## 6 Classifier Aggregation

The fuzzy measure [44] is a set function used to express the grade of fuzziness. Say, $\boldsymbol{X}$ is a non-empty universe of discourse and $B$, is a σ-field of $\boldsymbol{X}$ [14]. Given the sets, $A_1$ and $A_2$, $B$ is a family of subsets of $\boldsymbol{X}$ if: (i) $\emptyset \in B$; (ii) $\boldsymbol{X} \in B$; (iii) if $A_1 \in B$ then $\neg A_1 \in B$; and, (iv) $B$ is closed under set union (i.e., if $A_1 \in B$ and $A_2 \in B$ then $A_1 \cup A_2 \in B$)]. The set function, $g{:}B \rightarrow [0,1]$, is a fuzzy measure over $\boldsymbol{X}$ if three axioms hold: (i) $g(\emptyset)=0$ and $g(\boldsymbol{X})=1$ (boundary conditions ensure that regardless of the degree of evidence an element must not belong to the null set and it must belong to the universe of discourse); (ii) if $A_1, A_2 \in B$ and $A_1 \subset A_2$ then $g(A_1) \leq g(A_2)$ (evidence of an element's membership in a set must always be at least as great as that in any of the set's subsets); and, (iii) if $A_1 \in B$ and $A_1$ is monotone increasing then $\lim g(A_1)=g(\lim A_1)$ (consistency constraint). A fuzzy measure commonly found in the literature is the Sugeno fuzzy measure [45], $g_\lambda$, which satisfies the additional constraint that $g(A_1 \cup A_2)=g(A_1)+g(A_2)+\lambda g(A_1)g(A_2)$, where $\lambda>-1$ and $A_1 \cap A_2=\emptyset$.

The fuzzy integral [11] is a nonlinear aggregation scheme for combining multiple sources of information to arrive at a "confidence value" for a decision (hypothesis). Let us define a mapping $h{:}\boldsymbol{X} \rightarrow [0,1]$ where a finite ordered $\boldsymbol{X}=\{x_1 \ldots x_n\}$ is of interest. Typical examples are the Sugeno, $Su(x)$, Choquet, $Ch(x)$, and Shilkret, $Sh(x)$, integrals [5,19,20]. The fuzzy integrals of $h$ over $\boldsymbol{X}$ with respect to $g_\lambda$ are defined as:

$$\begin{aligned} Su(x) &= \bigvee_i \left[ h(x_i) \wedge g_\lambda(\boldsymbol{X}_i) \right] \\ Ch(x) &= \bigvee_i \left[ \left( h(x_i) - h(x_{i-1}) \right) g_\lambda(\boldsymbol{X}_i) \right] \\ Sh(x) &= \bigvee_i \left[ h(x_i) \cdot g_\lambda(\boldsymbol{X}_i) \right] \end{aligned} \tag{12}$$

where $\boldsymbol{X}_i=\{x_1, x_2, \ldots, x_i\}$ and $h(x_0)=0$. While several possible interpretations exist for the conceptual meaning of a fuzzy integral [46,48], in this discussion it is considered to mean the maximum degree of belief (for a prediction or classification outcome) obtained by the fusion (aggregation) of several sources of objective evidence.

Integrating the results from multiple classifiers involves using their respective confusion matrices to compute the fuzzy densities for each of the classifiers in order to determine the fuzzy measures used in (12). To this end, the technique described in [4] is followed primarily and is briefly described here. Let $\boldsymbol{R}_k=(n_{kij})$ be the $c \times c$ confusion matrix for classifier, $k$, where $n_{kii}$ is the number of class $i$ patterns that were correctly classified by $k$ and $n_{kij}$ $(i \neq j)$ is the number of class $i$ patterns that were incorrectly assigned to class $j$ by $k$. The preliminary fuzzy density of class $i$ with respect to classifier $k$, $0<g^*_{ki}<1$, is

$$g^*_{ki} = \frac{n_{kii}}{\sum_{j=1}^{c} n_{kij}} \tag{13}$$

These densities must be adjusted to take into account the frequencies of correct and incorrect classifications within and across the set of classifiers. This leads to the following expressions

$$\delta_{kij} = \begin{cases} 1 & i = j \\ \dfrac{n_{kii} - n_{kij}}{n_{kii}} & i \neq j \\ \varepsilon & n_{kii} < n_{kij} \end{cases}, \quad \gamma_{kij} = \begin{cases} 1 & n_{kij} < n_{lij} \\ \dfrac{n_{lij}}{n_{kij}} & n_{kij} \geq n_{lij} \\ \varepsilon & n_{kij} = 0 \end{cases} \tag{14}$$

where ε is a small positive constant. The corrected fuzzy density, $g_{ki}$, may now be computed as

$$g_{ki} = g_{ki}^{*} \times \left(\delta_{kir} \times \cdots \times \delta_{kis}\right)^{w_1} \times \left(\gamma_{kir} \times \cdots \times \gamma_{kis}\right)^{w_2} \tag{15}$$

where $w_1$ and $w_2$, ($w_1$+$w_2$=1) are weighting factors and $r$ and $s$ are the indices of those pattern classes for which classifier $k$ produced the highest classification accuracy. The first adjustment, $\delta \in (0,1]$, reflects the pattern misclassifications within the confusion matrix for $k$. As the pattern misclassifications increase, $\delta \rightarrow 0$ (the third condition in (14) represents the degenerate case when more patterns of a particular class are misclassified than correctly classified). The second adjustment, $\gamma \in (0,1]$, reflects the pattern misclassifications across all classifiers with respect to $k$. As the pattern misclassifications increase, $\gamma \rightarrow 0$ (the third condition in (14) is the degenerate case when no patterns of a particular class are correctly classified).

Finally, the Sugeno, Choquet, and Shilkret integrals can exploit several variants of $h$ including: $h_c(x)$, contrast intensification as defined by (11), and $h_p(x)=x^p$ ($p$>0), where $x \in [0,1]$ is the classifier's predicted class label assignment. When 0<$x$<1, $h(x)$ will act to dilate membership values, while concentration will occur when $x$>1. In order to constrain the number of parameters, the standard fuzzy set based definitions for concentration ($p$=2) and dilation ($p$=0.5) are normally used. In total, four variants are typical candidates for the integrals: $h_c(x)$, $h_{0.5}(x)$, $h_2(x)$, and $h_1(x)$ (identity). Finally, using equations (12)–(15), the actual class label output from the set of pattern classifiers is the one with the highest integrated value.

## 7 Experiments, Analysis and Results

In this concluding section, we present a series of experiments, which employed the classification approaches described above, relating to the interpretation, analysis, and classification of several biomedical datasets. A summary of the results, listed in Table 2, may be found at the end of this section.

### 7.1 FQE

In [34], MR spectra were obtained (360 MHz) for 25 thyroid biopsies: 16 papillary carcinomas and 9 normal. Two spectral regions were analyzed: the main lipid CH2 and CH3 peaks, 0.64–2.59 ppm; and the choline-like species, 2.59–3.41 ppm. Analysis was based on 170 features for the choline region and 400 features for the lipid region. As a benchmark, the inputs to an MLP classifier were the 10 principal components of the dataset that accounted for 97% of the cumulative variance [37]. FQE was used with 680 (choline) and 1600 (lipid) membership values.

FQE significantly outperformed the benchmark: $P_O$=0.92 versus $P_O$=0.64 (choline); $P_O$=0.88 versus $P_O$=0.80 (lipid). Of particular interest is the significant reduction in convergence rate for the FQE MLP, O($10^3$) versus O($10^6$) for the benchmark.

In [30], data were analyzed pertaining to tonsillectomy/adenoidectomy patients with predispositions to excessive bleeding. These blood abnormalities include hemophilia, a hereditary hemorrhagic diathesis due to coagulation cofactor FVIII deficiency; von Willebrand's disease, a diathesis associated with von Willebrand protein antigen factor deficiencies or in the activity measured as the restocetin cofactor; and thrombopathy, a platelet function defect measured as the occurrence of at least two abnormal platelet aggregation [21]. Data were collected from the patient database associated with a hematology expert system containing information relating to coagulation laboratory test results and patients responses to a bleeding tendency questionnaire.

Two major experiments were conducted. In the first, 96 patient records (patterns) were assigned to one of three disease states (class labels): 42 hemophilia (H), 30 platelet function defect (P), and 24 von Willebrand's disease (V). LDA, MLP, and FQE (with MLP) classifiers were used in the analysis: respectively, $\kappa$=0.55 (moderate agreement), $\kappa$=0.71 (substantial agreement), and $\kappa$=0.79 (substantial agreement). MLP and FQE also had consistently better classification results across all three disease states with particularly strong improvements with H and V. While FQE outperformed MLP with respect to correctly classifying P (80% versus 70%), it under performed with respect to V (83% versus 88%). However, FQE was clearly superior in classifying H; 93% versus 81%. FQE, on average, converged 4.2 times faster during the training phase than MLP.

In the second set of experiments, a different gold standard was used (derived from the expert system) to assign 191 patient records to either a normal (N) or abnormal (A) class. The records were randomly assigned to a design set (60 N and 60 A) or a validation set (42 N and 29 A). The respective $\kappa$ scores for LDA, MLP, and FQE were 0.16 (slight agreement), 0.39 (fair agreement), and 0.46 (moderate agreement).

In [32], dispersion-adjusted FQE (DFQ) MLP classifiers were used in the analysis and classification of three biomedical datasets found in the Machine Learning Repository (http://mlearn.ics.uci.edu/MLSummary.html) at the University of California, Irvine. The patterns in these three datasets belong to one of two possible classes: "target", where the pattern belongs to an abnormal or disease state; and, "control", where the pattern belongs to a normal or control state.

In the first case, the heart data [16] is a description of diagnoses relating to $N$=267 cardiac single proton emission computed tomography images [10]. The $n$=44 features relate to frequency information across 22 different regions of interest and alternate between images taken while the patient was at rest or during a controlled stress condition (target=55, control=22). The overall classification accuracy using the original features was $P_O$=0.80 while the FQE accuracy was $P_O$=0.92 and the DFQ accuracy was $P_O$=0.95 (a respective 15% and 19% increase in performance). DFQ decreased the false positive error rate from 10% to 7% with an overall increase in accuracy of 3%.

In the second case, each of the $N$=155 patterns (target=32, control=123) within the hepatitis dataset [2] is composed of 19 features: 6 nominal features (age, bilirubin, alkaline phosphate, SGOT, albumin, and protime) and 13 binary features (sex, steroids, antivirals, fatigue, malaise, anorexia, large liver, firm liver, palpable spleen, spiders, ascites, varices, and histology). The overall accuracy using the original features was $P_O$=0.88 while the FQE accuracy was $P_O$=0.91 and the DFQ accuracy was $P_O$=0.94 (a respective 3% and 7% increase in classifier performance). With FQE, this improvement was gained exclusively by a reduction in the false negative error rate (from 37% to 22%). In the DFQ case, a greater reduction in the false negative error rate (19%) was achieved with an overall increase in accuracy of 3% compared to the FQE encoding.

In the third case, the lung cancer data [12], which comprises 56 nominal features taking on integer values (0–3), represents three different types of pathological lung cancers. Due to the paucity of patterns ($N$=32) in this dataset, and in the interest of simplifying the comparative analysis with the other two biomedical datasets, the two classes with the fewest patterns are merged into one pathological (target) case (control=13, target=19). The overall classification accuracy using the original features was $P_O$=0.63 while the FQE accuracy was $P_O$=0.78 and the DFQ accuracy was $P_O$=0.84 (a respective 23% and 33% increase in classifier performance). DFQ achieved an 8% improvement in classification performance compared to FQE.

## 7.2 GSA

In [31], GSA was used in the analysis of a biomedical dataset composed of 206 $^1$H MR spectra (360 MHz, 37°) consisting of 95 meningiomas (M), 74 astrocytomas (A), and 37 control samples of non-tumorous brain tissue from patients with epilepsy (E). The biomedical spectra ($n$=550 in the region of 0.3–4.0 ppm) were randomly assigned to either a design ($N_D$=80, with 29 M, 31 A, and 20 E) or a validation set ($N_V$=126). Applying GSA to the gold standard (provided by a pathologist) improved the overall diagnostic (classification) performance of an MLP classifier by 13%: $\kappa$=0.80 versus $\kappa$=0.71 using the original design class labels.

Although none of the spectra (patterns) in the validation set was reclassified, two validation spectra were flagged as outliers (two M spectra were flagged as A), and these spectra were indeed misclassified as A. Classification errors were also more conservative. Using the original class labels, 5 E's (control) were classified as tumors (M or A) and 4 tumors as control. However, in the case of GSA, only 1 E was misclassified as a tumor while only 3 tumors were misclassified as control.

## 7.3 SFS

In [33], SFS was used in the analysis and classification of $N$=444 $^1$H MR spectra (360 MHz at 37°C) of isolates of five different species of Candida yeast ($n$=1500): 104 albicans (A), 93 parapsilosis (P), 81 krusei (K), 75 tropicalis (T), and 91 glabrata (G). The design set comprised 50 randomly selected patterns from each class. The feature region cardinality range was 7–1231.

The mean accuracy (for the 10,000 MLP processes) was $P_O$=0.83 for the validation set. The best accuracy score was $P_O$=0.95 for an MLP using only 16 of the 1500 features (~1%). Interestingly, the top four accuracy scores were achieved by MLPs that used less than 20 features.

In [29], SFS was used in the classification of two biomedical datasets. In the first case, 186 infrared spectra of synovial joint fluid were assigned to one of three disease states: 72 rheumatoid arthritis (R), 72 osteo-arthritis (O), and 42 control samples (C). The spectra ($n$=2801) cover the wavelength range 1000–3700 cm$^{-1}$. The pattern design set contained 28 randomly selected spectra from each class. The feature region cardinality range was 7–192 and the mean κ score was 0.79 for the 20,000 MLP and PNN classification processes. The best validation κ score was 0.86±0.02 (almost perfect agreement) using the MLP classifier with only 20 of the 2801 original features (<1%). The best validation κ score for PNN was 0.84±0.02 (almost perfect agreement) using 25 of the original features (<1%). Using the original 2801 spectral features (i.e., no stochastic feature selection), the PNN benchmark produced a validation set κ score of only 0.51 (moderate agreement), while the MLP benchmark κ score was only 0.29 (fair agreement).

In the second case, this is likely due to over-fitting of the design (training) data as is evidenced by the high κ score of 0.88. Due to the inversion of the large covariance matrix, LDA produced spurious results. Next, the original infrared spectra were averaged down to 100 features. All three benchmarks performed well (substantial agreement): PNN, κ=0.69; MLP, κ=0.74; LDA, κ=0.69. While slightly worse than the average of all 10000 PNN (and MLP) runs using SFS, they were appreciably worse than the best runs. In the second case, 227 MR spectra of a biological fluid discretized ($n$=512) were assigned to one of three classes: 108 normal (N), 54 of borderline character (B), and 65 abnormal (A). The design set contained 36 randomly selected samples from each class.

The feature region cardinality range was 4–212. For the 10000 MLP classification processes, the mean κ score was 0.42 for the validation set. For the 10000 PNN classification processes, the mean κ score was 0.48 for the validation set. The best validation κ score was 0.54±0.02 (moderate agreement) using MLP and 83 of the 512 original features (16%). The best validation κ score for PNN was 0.48±0.02 (moderate agreement) using 79 of the original features (15%). Using the original 512 spectral features (again no feature selection), all benchmarks performed poorly (only fair agreement): PNN, κ=0.38; MLP, κ=0.25; LDA, κ=0.24. As with the infrared spectra, the MLP likely over-fitted the design data (κ=0.88). Finally, the original dataset was averaged down to 128 features. All benchmarks had moderate levels of agreement: PNN, κ=0.47; MLP, κ=0.46; LDA, κ=0.43.

## 7.4 Classifier Aggregation

In [35], fuzzy aggregation was used in conjunction with SFS in the analysis and classification of $N$=191 MR spectra ($n$=3380) of a biofluid that were assigned to one of two classes by a medical expert: 116 normal and 75 abnormal. The design set comprised 58 randomly selected patterns from each class. Three transformed dataset variants were generated: first derivative; rank ordered; and first derivative with rank ordering. The best validation set result was $P_O$=0.79 using the fuzzy aggregation approach with rank ordered transformed features, which is an 8% improvement over the corresponding best individual (PNN) classifier. Further, the aggregated approach outperformed the corresponding best individual classifiers across all variants: respectively ($P_O$), 0.76/0.74, 0.74/0.62, 0.79/0.73, 0.75/0.73.

**Table 2.** Summary of biomedical data classification results

| Description | N | n | P | Method | Benchmark |
|---|---:|---:|:---:|:---:|:---:|
| FQE: MRS Thyroid I | 25 | 170 | $P_O$ | 0.92 | 0.64 |
| FQE: MRS Thyroid II | 25 | 400 | $P_O$ | 0.88 | 0.80 |
| FQE: Hematology I | 96 | 11 | $\kappa$ | 0.79 | 0.71 |
| FQE: Hematology II | 191 | 9 | $\kappa$ | 0.46 | 0.39 |
| DFQ: Heart | 267 | 44 | $P_O$ | 0.95 | 0.92 |
| DFQ: Hepatitis | 155 | 19 | $P_O$ | 0.94 | 0.91 |
| DFQ: Lung Cancer | 32 | 56 | $P_O$ | 0.84 | 0.78 |
| GSA: MRS Brain | 206 | 550 | $\kappa$ | 0.80 | 0.71 |
| SFS: MRS Candida | 444 | 1500 | $P_O$ | 0.95 | 0.83 |
| SFS: Synovial Fluid | 186 | 2801 | $\kappa$ | 0.86 | 0.74 |
| SFS: MRS Biofluid | 227 | 512 | $\kappa$ | 0.54 | 0.47 |
| Fusion: MRS Biofluid | 191 | 3380 | $P_O$ | 0.79 | 0.73 |

Each entry lists the biomedical classification method examined (as described in this section), the dataset used in the evaluation, the number of patterns (N), the number of features (n), the performance measure (P), the method's overall accuracy, and the accuracy for the best benchmark.

## 8 Conclusion

The analysis, interpretation, and classification of biomedical data are replete with pattern recognition challenges stemming from the curse of dimensionality and tarnished gold standards. This chapter presents a computational intelligence based methodology, which remediates these challenges, exploiting strategies and methods inspired by the granular computing paradigm. Stochastic feature selection, gold standard class label adjustment, classifier aggregation, and fuzzy quantile encoding may be used singly or in concert within a classification system. As pre- and post-processing approaches, they may easily be incorporated into investigators' classifiers of choice.

**Acknowledgments.** We thank Conrad Wiebe and Aleksander Demko for implementing the stochastic feature selection algorithm and associated libraries. The following researchers are gratefully acknowledged for making their respective datasets publicly available for use by their peers: K.J. Cios and L.A. Kurgan for the SPECTF heart data; G. Gong and B. Cestnik for the hepatitis data; S. Aeberhard for the lung cancer data; and, U. Himmelreich for the yeast data. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

1. Bishop, C.M.: Neural networks and their applications. Rev. Sci. Instrum. 65, 1803–1832 (1994)
2. Cestnik, B., Kononenko, I., Bratko, I.: ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In: Bratko, I., Lavrac, N. (eds.) Progress in Machine Learning. Sigma Press, Wilmslow (1987)

3. Cheng, B., Titterington, D.M.: Neural networks: a review from a statistical perspective. Stat. Sci. 9, 2–54 (1994)
4. Chi, Z., Yan, H., Pham, T.: Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition. World Scientific, New Jersey (1996)
5. Choquet, G.: Theory of capacities. Annales de l'Institut Fourier 5, 131–295 (1953)
6. Demko, A.B., Pizzi, N.J., Somorjai, R.L.: Scopira – A system for the analysis of biomedical data. In: Proc. IEEE Can. Conf. Electr. Comput. Eng., Winnipeg, Canada, May 12–15, 2002, pp. 1093–1098 (2002)
7. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley Interscience, New York (2000)
8. Everitt, B.S.: Moments of the statistics kappa and weighted kappa. Br. J. Math. Stat. Psychol. 21, 97–103 (1968)
9. Fleiss, J.L.: Measuring agreement between judges on the presence or absence of a trait. Biom. 31, 651–659 (1975)
10. Friebolin, H.: Basic One- and Two-Dimensional NMR Spectroscopy. Wiley & Sons, New York (1998)
11. Grabish, M., Murofushi, T., Sugeno, M.: Fuzzy measure of fuzzy events defined by fuzzy integrals. Fuzzy Sets Syst. 50, 293–313 (1992)
12. Hong, Z.Q., Yang, J.Y.: Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognit. 24, 317–324 (1991)
13. Huber, P.J.: Robust estimation of a location parameter. Ann. Math. Stat. 35, 73–101 (1964)
14. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainty, and Information. Prentice-Hall, Englewood Cliffs (1988)
15. Kuncheva, L.I., Steimann, F.: Fuzzy diagnosis. Artif. Intell. Med. 16, 121–128 (1999)
16. Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M., Goodenday, L.S.: Knowledge discovery approach to automated cardiac SPECT diagnosis. Artif. Intell. Med. 23, 149–169 (2001)
17. Landis, J.R., Koch, G.G.: The measurements of observer agreement for categorical data. Biom. 33, 159–174 (1997)
18. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data Min. Knowl. Discovery 6, 393–423 (2002)
19. Mesiar, R., Mesiarová, A.: Fuzzy Integrals—What Are They? Int. J. Intell. Syst. 23, 199–212 (2008)
20. Murofushi, T., Sugeno, M.: An interpretation of fuzzy measure and the Choquet integral as an integral with respect to a fuzzy measure. Fuzzy Sets Syst. 29, 201–227 (1989)
21. Nosek-Cenkowska, B., Cheang, M.S., Pizzi, N.J., Israels, E.D., Gerrard, J.M.: Bleeding/bruising symptomatology in children with and without bleeding disorders. Thromb. Haemost. 65, 237–241 (1991)
22. Parzen, E.: On estimation of a probability density function and mode. Ann. Math. Stat. 33, 1065–1076 (1962)
23. Pavia, D.L., Lampman, G.M., Kriz, G.S.: Introduction to Spectroscopy. Harcourt Brace College, Fort Worth (1996)
24. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. 177, 3–27 (2007)
25. Pedrycz, W.: Granular Computing: The Emerging Paradigm. J. Uncertain Syst. 1, 38–61 (2007)
26. Pedrycz, W.: Granular Computing: An Emerging Paradigm. Springer, Heidelberg (2001)
27. Pedrycz, W., Gomide, F.: Fuzzy Systems Engineering: Toward Human-Centric Computing. Wiley & Sons, New York (2007)
28. Pizzi, N.J.: Biomedical data analysis using dispersion-adjusted fuzzy quantile encoding. In: Proc. Annu Meet North Am. Fuzzy Inf. Process Soc., New York, USA, #50010 (6 pages), May 19–22 (2008)
29. Pizzi, N.J.: Classification of biomedical spectra using stochastic feature selection. Neural Netw. World 15, 257–268 (2005)

30. Pizzi, N.J.: Bleeding predisposition assessments in tonsillectomy/adenoidectomy patients using fuzzy interquartile encoded neural networks. Artif. Intell. Med. 21, 65–90 (2001)
31. Pizzi, N.J.: Fuzzy preprocessing of gold standards as applied to biomedical spectra classification. Artif. Intell. Med. 16, 171–182 (1999)
32. Pizzi, N.J., Pedrycz, W.: An analysis of potentially imprecise class labels using a fuzzy similarity measure. In: Proc. World Congr. Comput. Intell., Hong Kong, June 1–6, pp. 667–672 (2008)
33. Pizzi, N.J., Pedrycz, W.: Classification of magnetic resonance spectra using parallel randomized feature selection. In: Proc. Int. Jt. Conf. Neural Netw., Budapest, Hungary, July 25–29, 2004, pp. 2455–2460 (2004)
34. Pizzi, N., Somorjai, R.L.: Fuzzy encoding as a preprocessing method for artificial neural networks. In: Proc. World Congr. Neural Netw., San Diego, USA, June 5-9, 1994, pp. 643–648 (1994)
35. Pizzi, N.J., Wiebe, C., Pedrycz, W.: Biomedical spectral classification using stochastic feature selection and fuzzy aggregation. In: Proc. Ann. Meet North Am. Fuzzy Inf. Process Soc., San Diego, USA, June 24–27, 2007, pp. 360–365 (2007)
36. Pizzi, N.J., Alexiuk, M.D., Pedrycz, W.: Classification of biomedical spectra using fuzzy interquartile encoding and stochastic feature selection. In: Proc. IEEE Symp. Ser. Comput. Intell. Data Min., Honolulu, USA, June 1–6, pp. 668–673 (2007)
37. Pizzi, N., Choo, L.-P., Mansfield, J., Jackson, M., Halliday, W.C., Mantsch, H.H., Somorjai, R.L.: Neural network classification of infrared spectra of control and Alzheimer's diseased tissue. Artif. Intell. Med. 7, 67–79 (1995)
38. Ripley, B.D.: Neural networks and related methods for classification. J. Royal Stat. Soc. [B] 56, 409–456 (1994)
39. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing, vol. 1. MIT Press, Cambridge (1986)
40. Seber, G.: Multivariate Observations. Wiley & Sons, New York (1984)
41. Snir, M., Gropp, W.: MPI: The Complete Reference. MIT Press, Cambridge (1998)
42. Somorjai, R.L., Dolenko, B., Nikulin, A.K., Pizzi, N., Scarth, G., Zhilkin, P., Halliday, W., Fewer, D., Hill, N., Ross, I., West, M., Smith, I.C.P., Donnelly, S.M., Kuesel, A.C., Briere, K.M.: Classification of 1H MR spectra of human brain neoplasms: The influence of preprocessing and computerized consensus diagnosis on classification accuracy. J. Magn. Reson. Imaging 6, 437–444 (1996)
43. Specht, D.F.: Probabilistic neural networks. Neural Netw. 3, 109–118 (1990)
44. Sugeno, A.: Fuzzy measures and fuzzy integrals: A survey. In: Gupta, M.M., Saridis, G.N., Gaines, B.R. (eds.) Fuzzy Automata and Decision Processes, pp. 90–102. North Holland, Amsterdam (1977)
45. Sugeno, A.: Theory of fuzzy integral and its applications. PhD Thesis, Tokyo Institute of Technology (1972)
46. Tahani, H., Keller, J.M.: Information fusion in computer vision using the fuzzy integral. IEEE Trans. Syst. Man Cybern. 20, 733–741 (1990)
47. Vapnik, V.: Statistical Learning Theory. Wiley & Sons, New York (1998)
48. Weiss, S.M., Kulikowski, C.A.: Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems. Morgan Kaufmann, San Mateo (1991)
49. Witten, I.H., Eibe, F.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Mateo (2005)
50. Zadeh, L.A.: Fuzzy logic = Computing with words. IEEE Trans. Fuzzy Syst. 4, 103–111 (1996)
51. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Man Cybern. 3, 28–44 (1973)

# Gene Interactions Sub-networks and Soft Computing

Ranajit Das and Sushmita Mitra

Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India
ranajit_r@isical.ac.in, sushmita@isical.ac.in

**Abstract.** Analysis of gene interaction networks is crucial for understanding fundamental cellular processes involving growth, development, hormone secretion and cellular communication. A gene interaction network comprises of proteins and genes binding to each other, and acting as a complex input-output system for controlling cellular functions. A small set of genes take part in a cellular process of interest, while a single gene may be involved in more than one cellular process at the same time. Soft computing is a consortium of methodologies that works synergistically and provides flexible information processing capability for handling real life ambiguous situations. The tools include fuzzy sets, evolutionary computing, neurocomputing, and their hybridizations. We discuss some existing literature pertaining to the use of soft computing and other classical methodologies in the reverse engineering of gene interaction networks. As a case study we describe here a soft computing based strategy for biclustering and the use of rank correlation, for extracting rank correlated gene interaction sub-networks from microarray data. Experimental results on time series gene expression data from *Yeast* were biologically validated based on standard databases and information from literature.

**Keywords:** Soft Computing, bioinformatics, multi-objective evolutionary biclustering, transcriptional regulatory network extraction, gene expression profile, rank correlation, gene interaction network.

## 1 Introduction

With the current development in microarray technology (gene chips), today researchers in Bioinformatics have, at their disposal, expression data of thousand of genes of different organisms under various experimental conditions. This had led to complete-genome expression profiling of several organisms. The latest Affymetrix gene chips contain 750,000 unique 25-mer oligonucleotide features constituting more than 28,000 mouse gene-level probe sets. This DNA microarray technology forms an indispensable tool for exploring transcriptional regulatory networks from the system level and is useful when one dwells into the cellular environment to investigate various complex interactions [1]. Biological networks connect genes, gene products (in the form of protein complexes) or their groups to one another. A network of co-regulated genes may form gene clusters that can encode proteins, which interact amongst themselves and take part in common biological processes. Clustering of gene expression profiles have been employed to identify co-expressed groups of genes [2] as well as to extract gene interaction/gene regulatory networks [3].

Sharing of the regulatory mechanism amongst genes, in an organism, is predominantly responsible for their co-expression. Genes with similar expression profiles are

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 313–327.
springerlink.com

very likely to be regulators of one another or be regulated by some other common parent gene [4]. Often, it is noted that during few conditions a small set of genes are co-regulated and co-expressed, their behavior being almost independent for rest of the conditions. The genes share local rather than global similar patterns in their gene expression profiles. Generally, group of genes are identified in the form of biclusters using continuous columns biclustering because biological processes start and terminate over a continuous interval of time [5, 6]. The aim of biclustering is to bring out such local structure inherent in the gene expression data matrix. It refers to the clustering of both rows (genes) and columns (conditions) of a data matrix (gene expression matrix), simultaneously, during knowledge discovery about local patterns from microarray data [7].

The genome, comprising the set of all genes in an organism along with their expressions values, is considered to be a switching network, with its vertices denoting the proteins or molecules and the directed edges representing their various interactions and inter-dependence. Such networks relate genes, gene products or their groups (like protein complexes or protein families) to each other. A directed edge (or arc) connects one node (or vertex) to another. Consider the graph depicted in Fig. 1. Mathematically a network can be expressed as a graph $G = \{V, E\}$, where $V$ represents the set of $N$ vertices $\{V_1, V_2, ..., V_N\}$ while $E$ represents the set of edges that connect two elements in $V$.

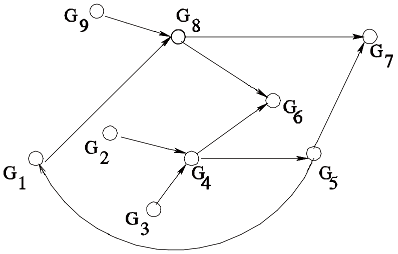


**Fig. 1.** A sample gene interaction network with nine nodes and ten edges

In this chapter we provide an overview on the extraction of gene interaction networks followed by a study involving a rank correlation-based multi-objective evolutionary technique for the extraction of simple gene interaction sub-networks from microarray data. Use of soft computing, with biclustering, is described in this connection. Preprocessing, involving the discretization of the rank correlation matrix (using quantile partitioning) and subsequent elimination of weak correlation links, is employed to retain strongly rank correlated (positive or negative) gene interaction pairs. Experimental results on *Yeast* data are validated in terms of a gene ontology (GO) study. The rest of the chapter is organized as follows. Section 2 introduces the basics of biological networks and gene interaction networks. Section 3 describes some classical reverse engineering approaches for generating gene interaction networks using time series gene expression data. In Section 4, the existing literature pertaining to the use of soft computing in the extraction of gene interaction networks is compiled. As a case study, the use of multi-objective evolutionary biclustering and rank correlation for the extraction of Gene interaction sub-network is described in Section 5. The effectiveness of the discussed methodology is also demonstrated therein, using time-series gene expression data from *Yeast*. The article is concluded in Section 6.

## 2 Biological Networks

Biological pathways can be conveniently represented as networks and broadly classified as *metabolic pathways*, *signal transduction pathways* and *gene interaction networks*. The repository of information about various biological pathway data is available in some databases like BioCyc[1] [8], EcoCyc [9], What Is There (WIT) system[2], RegulonDB [10], *etc*. GO [11] and the KEGG Orthology [12] promote the use of controlled vocabulary to facilitate computational analysis. These databases can be integrated with various computational methods to get an insight into complex biological functions. They can help in (i) reconstructing biochemical pathways from the complete genome sequence, and (ii) predicting gene interaction networks. The proper understanding of gene interaction networks is essential for the understanding of fundamental cellular processes involving growth and decay, development, secretion of hormones, cellular communication, etc. During transcription of gene expression specific groups of genes may be made active by certain signals, which on activation, may regulate similar biological processes. The genes may also be regulators of each other’s transcription.

The metabolic pathways facilitate mass generation, energy production, information transfer and cell-fate specification, in a cell or micro-organism; they are seamlessly integrated through a complex network of cellular constituents and reactions. Such a metabolic network consists of nodes, *i.e.*, substrates (genes or proteins), which are interconnected through links, *i.e.*, metabolic reactions in which enzymes provide the catalytic scaffolds [13].

Signal transduction is the process by which a cell converts one kind of signal or stimulus into another by a series of steps, causing functional changes inside the cell. The signal may pass from one cell to another (Hormone-Receptor concept), from extracellular environment to inside the cell (through plasma membrane) or from one compartment inside the cell to another compartment (*i.e.*, from cytoplasm to nucleus). A signal transduction pathway can be considered as a biological network of biomolecules connected by various kinds of interactions (protein-protein interactions, protein-ion interactions, *etc.*) among them.

Analyzing various types of messenger RNAs (mRNAs) produced by a cell and quantifying them, one can determine the gene or set of genes that get transcribed under particular experimental conditions. A cell dynamically responds to both environmental stimuli and its own changing requirements in a highly complicated and tightly regulated process. This process helps one to monitor the required increase or decrease of the expression levels of particular genes. The control and regulation of gene expression could be caused by various external factors, occurring at different stages of

[1] http://www.biocyc.org/

[2] http://wit.integratedgenomics.com/

the cellular information flow from DNA, RNA to protein, like in mRNA splicing, translational control and/or post-translational control. Nevertheless, the one involving the initiation of transcription has been most widely studied in literature [14, 15, 16].

A gene regulatory network (GRN) determines which subset of genes is expressed, up to what level, and in response to what conditions of the cellular environment. While the metabolic networks form the basis for the net accumulation of biomolecules in living organisms, the regulatory networks modulate their action – thereby leading to physiological and morphological changes. However, one should note that any apparent similarity of expression profiles between two genes may not always mean that they may regulate each other but may signify (i) indirect co-regulation by other genes, (ii) direct regulation of one gene by the other, or (iii) a mere coincidence involving no causal relationship. An integration of additional biologically relevant knowledge may, therefore, provide constraints on suitable identification of groups of co-regulated genes.

## 3 Reverse Engineering of Genetic Interaction Networks

Reconstruction of interactions in gene regulatory networks, from gene expression data, is termed reverse engineering. Some of the techniques, typically used for the purpose, include the generalized Bayesian networks [17, 18], Boolean networks [19, 20, 21, 22], linear and non-linear ordinary differential equations (ODEs) [14, 23, 24]. Boolean networks are binary models with genes taking on values one (or zero) to represent active (or inactive) states [22]. However these ignore the effect of genes at intermediate levels, and result in information loss during discretization. Bayesian networks are graph models that estimate complicated multivariate joint probability distributions through local probabilities [17]. Reverse engineering with Bayesian learning [18] enabled the generation of gene regulatory interactions from simulated gene expression data. Dynamic Bayesian networks (DBNs) were subsequently used for inferring the relationship amongst genes from time-series gene expression data [25, 26].

Gene regulatory relationships were extracted for cell cycle-regulated genes in *yeast*, with the activation or inhibition between gene pairs being represented as events [27]. Matching of corresponding events was followed by a sequence alignment of the event strings. Regulatory relationships have also been deduced from the correlation of co-expressions, between a DNA-binding transcription regulator and its target gene, by using a probabilistic expression model [28]. However, correlation matching alone is deemed unsuitable to effectively distinguish between regulators and target genes. It is also difficult to discern whether the correlated target is directly or indirectly regulated. Hence additional information like protein-DNA binding has been integrated into transcriptional regulatory networks [29] for validating direct regulator-target interaction.

Co-regulated genes are often functionally, *i.e.*, physically, spatially and/or genetically associated. In real life, however, the genes may be co-regulated only across a subset of all observed experimental conditions. In other words, a small number of genes participate in a cellular process of interest while a gene may be simultaneously active in more than one cellular process. It is here, where biclustering (or coclustering) becomes more appropriate than standard clustering, for the purpose of modeling regulatory pathways. Here we perform simultaneous clustering of both rows (genes) and columns (conditions) of the gene expression matrix, for knowledge discovery in maximal subgroups of local patterns [30, 31]. An algorithm *cMonkey* has been developed [32], to detect putatively co-regulated gene groupings by integrating biclustering of gene expressions and various functional associations with the *de novo* detection of sequence motifs.

## 4 Role of Soft Computing

In addition to the combinatorial approach, soft computing is gradually opening up several possibilities by generating low-cost (computational cost both in terms of space and time complexity), low-precision (approximate), good solutions. Soft computing is a consortium of methodologies that works synergistically and provides flexible information processing capability for handling real life ambiguous situations [33]. The tools include fuzzy sets, evolutionary computing, neurocomputing, and their hybridizations. Typically, they require little a priori knowledge about the underlying system, and the model can be derived directly from the data. Since the work deals with huge amounts of incomplete or ambiguous data, (i) the uncertainty handling capacity of fuzzy sets, (ii) the learning ability of artificial neural networks (ANNs) to discover hidden regularities within the data, and (iii) the searching potential of evolutionary strategies (like genetic algorithms) to explore the large pattern space, are typically utilized [34].

The human mind expresses higher level of perceptions using vague, non-crisp concepts. So for developing really intelligent methods for approximate reasoning about similar concepts accessible for intelligent systems, languages need to be developed. One way out while searching for solutions to these tasks is the use of Granular Computing. Granular computing [35] (GC) is useful in finding meaningful patterns in data by expressing and processing chunks of information (granules). The solutions involving GC become feasible because they specify non-Boolean or non-crisp specifications to a satisfactory degree and can be, more often than not, efficiently constructed than those involving detailed, purely numeric solutions. GC may thus be informally defined as a general computing theory for effectively using granules in the form of classes, clusters, subsets or groups, *etc.* and intervals for developing efficient computational models for complex applications involving huge amount of data, information and knowledge [36].

A problem that we conceive of is generally cast into frameworks, which facilitate the observations about clusters of objects with some commonality and eventually lead to the effective formulation of the problem and its solution with considerable acuity [35]. Such frameworks are ideal for problems involving pattern recognition, feature selection and reduction, knowledge discovery and bioinformatics. Identification of relevant features of objects contained in information granules help us to formulate hypotheses about the significance of the objects, construct new granules and refine the information, use GC to measure the distance among complex granules, *etc.* GC brings together the existing formalisms of set theory, fuzzy sets, and rough sets under a common platform by clearly visualizing some fundamental similarities and synergies.

The modeling of imprecise and qualitative knowledge, as well as the transmission and handling of uncertainty at various stages are possible through the use of fuzzy sets. Fuzzy logic is capable of supporting, to a reasonable extent, human type reasoning in natural form. Fuzzy Adaptive Resonance Theory (FART) associated matrix method has been developed [37] to cluster gene expression profiles of *Saccharomyces cerevisiae* (yeast) responding under oxidative stresses, followed by the extraction of genetic networks from them. The inferred genetic interactions are quantitatively evaluated, and validated in terms of the KEGG metabolic map, BRITE[3] protein interaction map and related literature. The number of clusters is controlled by the vigilance parameter of FART. Fuzzy rules of an activator-repressor model of gene interactions were used [38] to transform expression values into qualitative descriptors. A new multiscale fuzzy c-means clustering method was designed to model gene interactions between regulatory pathways, across different conditions and at different levels of detail [39].

The adaptivity of artificial neural networks (ANNs) to learn from data-rich environments and their robustness to noise make them good candidates for modeling genetic interactions from gene expressions. Some such connectionist models employed for extracting genetic regulatory effects include perceptrons [40, 41], self-organizing maps [42, 43], and recurrent neural networks (RNNs) [44, 45]. The RNN was used to model the dynamics of gene expression in the *lambda phage*[4] regulatory system [44].

Use of genetic algorithm (GAs) for reconstructing genetic networks has been reported in literature [46, 47]. The mutation and crossover operators help to intelligently guide the GA in the complex search space. Typically the GA searches for the most likely genetic networks that best fit the data, considering the set of genes to be included in the network along with the strength of their interactions. Gene interaction networks were inferred from microarray data [48], using GAs for interactive reverse engineering. However the combinatorial complexity is expected to be unmanageable in real-world problems, involving a large number of genes [49].

Hybrid techniques like neuro-fuzzy computing have found applications in the realm of genetic networks as well. ANNs and fuzzy logic have been employed to form a framework for inferring gene interaction networks. Knowledge-based neural networks, which incorporated prior knowledge about gene interactions, were used by Kasabov [50] for the reverse engineering of genetic networks. A hybrid methodology for this purpose has been developed [51] by combining ANN, fuzzy sets and multi-objective GAs.

## 5 Extraction of Gene Interaction Network: A Multi-objective Evolutionary Approach

Biological networks involving gene pairs, which demonstrate transcription factor (TF)-target relationship, is an important research problem. A gene interaction network is a complex structure comprising various gene products activating or repressing other gene products. A gene that regulates other genes is termed the transcription factor,

[3] KEGG BRITE Database is a collection of hierarchical classifications representing knowledge on various aspects of biological systems. http://www.genome.jp/kegg/brite.html

[4] Enterobacteria phage λ (lambda phage) is a temperate bacteriophage that infects the bacteria *Escherichia coli.*

while the gene being regulated is called its target. The presence of a TF, can alternatively switch “ON” some genes in the network while others remain “OFF”, orchestrating many genes simultaneously. The proper understanding of gene interaction networks is essential for the understanding of fundamental cellular processes involving growth and decay, development, secretion of hormones, *etc.* During transcription of gene expression specific groups of genes may be made active by certain signals, which on activation may regulate similar biological processes. The genes may also be regulators of each other’s transcription. Target genes sharing common TFs demonstrate similar gene expression patterns along time [14, 52]. Analysis of similar expression profiles brings out several complex relationships between co-regulated gene pairs, including co-expression, time shifted, and inverted relationships [53].

We describe a methodology for modeling the relationship between a transcription factor and its target’s expression level variation over time in the framework of the generated biclusters. The extraction of the relationship between the gene pair is biologically more meaningful and computationally less expensive as a bicluster is a subset of highly correlated genes and conditions. Rank correlation provides a similarity measure, which retains the relevant information necessary for computing pairwise correlation between gene pairs. The relationship is presented in terms of rules, where a TF is connected to its regulated target gene. These rules are subsequently mapped to generate parts of the entire regulatory network. It may be noted that intra-pathway gene interactions, responsible for a particular biological function and possibly within a bicluster, are generally stronger than any inter-pathway interactions.

The goal in genetic networks is to identify possible direct excitatory and/or inhibitory connections between genes, gene products and proteins, when the time-steps are close enough. Otherwise, indirect connections, through a third gene, needs to be established. Sometimes additional biological knowledge, such as gene ontology[5] and transcription factors, is included.

Most real-world search and optimization problems typically involve multiple objectives. A solution that is better with respect to one objective requires a compromise in other objectives. In problems with more than one conflicting objective there exists no single optimum solution. Rather, there exists a set of solutions, which are all optimal involving trade-offs between conflicting objectives. Unlike single-objective optimization problems, the multi-objective evolutionary algorithms (MOEA) tries to optimize two or more conflicting characteristics represented by fitness functions. Modeling this situation with single-objective GA would amount to heuristic determination of a number of parameters involved in expressing such a scalar-combination-type fitness function. MOEA, on the other hand, generates a set of Pareto-optimal solutions, which simultaneously optimize the conflicting requirements of the multiple fitness functions. Among the different multi-objective algorithms, it is observed that non-dominated sorting genetic algorithm (NSGA-II) possesses all the features required for a good MOEA. It has been shown that this can converge to the global Pareto front, while simultaneously maintaining the diversity of population. More details on the characteristics of NSGA-II, like non-domination, crowding distance and crowding selection operator can be found in [54].

[5] A shared, controlled vocabulary that is being developed to cover all organisms, in terms of molecular function, biological process and cellular component. http://www.geneontology.org

Biclustering refers to the simultaneous clustering and redundant feature reduction involving both attributes and samples. This results in the extraction of biologically more meaningful, less sparse partitions from high-dimensional data, and exhibit similar characteristics. The partitions are known as biclusters. Biclustering has been applied to gene expressions from cancerous tissues [31], mainly for identifying co-regulated genes, gene functional annotation, and sample classification. A bicluster can be defined as a pair $(g, c)$, where $g \subseteq \{1, ..., m\}$ represents a subset of genes and $c \subseteq \{1, ..., n\}$ represents a subset of conditions (or time points). The optimization task [30] involves finding the maximum-sized bicluster not exceeding a certain homogeneity constraint mentioned below. The size (or volume) f(g, c) of a bicluster is defined as the number of cells in the gene expression matrix $E$ (with values $e_{ij}$ ) that are covered by it. The homogeneity $G(g, c)$ is expressed as a mean squared residue score. More details on the biclustering scheme can be obtained in [54].

The Multi-objective GA (NSGA II), in association with the local search procedure discussed in [54], was used for the generation of the set of biclusters. The algorithm followed is discussed in details in [54]. The maximal set of genes and conditions representing size were generated keeping the "homogeneity" criteria of the biclusters intact. Since these two characteristics of biclusters are conflicting to each other, multi-objective optimization was employed to model them. To optimize this conflicting pair, the fitness function $f_1$ (corresponding to size) is always maximized while function $f_2$ (reflecting ratio of means square residual error and the threshold) is maximized as long as the residue is below the threshold, $\delta$ .

Like GC biclustering also contains some condensed information pertaining to correlation/co-regulation among subset(s) of genes. So, this helps in the extraction of gene interaction sub-networks, which appear to be more understandable to the human end-user.

### 5.1 Correlation between Gene Pairs

In this section we demonstrate the efficacy of a rank correlation-based approach for the extraction of gene interaction networks. A small number of genes participate in a cellular process of interest, being expressed over few conditions. Co-regulated genes are often found to have similar patterns in their gene expression profiles locally, rather than globally. The genes share similar sub-profiles, over a few time points, instead of the complete gene expression profiles. Thus, considering the global correlation amongst genes, *i.e.*, computation of correlation amongst genes employing the complete gene expression data matrix, would not reveal proper relationship between two of them. The *Spearman rank correlation* provides such a local similarity measure between the two time-series curves, since it is shape-based. The expression profile $e$ of a gene may be represented over a series of $n$ time points. Since the genes in a bicluster are co-expressed, the concept of correlation has been used to quantify their similarity. Instead of the commonly used similarity measures like the Euclidean distance or the Pearson correlation the *Spearman rank correlation* ( $RC$ ) have been employed due to its robustness towards outliers and measurement errors [55, 56].

Moreover, $RC$ does not assume a Gaussian distribution of points. $RC(e_1, e_2)$ between gene expression profile pair $e_1$ and $e_2$ provides a shape-based similarity measure between the two time-series curves, sampled at $e_{1i}$ and $e_{2i}$ over $n$ time intervals. This is expressed as

$$\mathrm{RC}(\mathrm{e}_1, \mathrm{e}_2) = 1 - \frac{6}{\mathrm{n}(\mathrm{n}^2 - 1)} \sum_{\mathrm{i}} [\mathrm{r}_{\mathrm{e}1}(\mathrm{e}_{1\mathrm{i}}) - \mathrm{r}_{\mathrm{e}2}(\mathrm{e}_{2\mathrm{i}})]^2, \qquad (1)$$

where $r_{e1}(e_{1i})$ is the rank of $e_{1i}$. Here an extended version of the RC has been used which takes into account the resolving of ties, *i.e.,* $e_{1j} = e_{1i}$ for $i \neq j$. The $RC$ satisfies $-1 \leq RC(e_1, e_2) \leq 1$ for all $e_1$, $e_2$.

The first preprocessing step is to filter correlation coefficients, which contribute minimally towards regulation. This is because often an exhaustive search of the possible interactions between genes is intractable. Next those coefficients are selected whose absolute values are above a detection threshold, suggesting greater correlation amongst the gene pairs. In this way we focus on a few highly connected genes that possibly link the remaining sparsely connected genes. The correlation range $[RC_{\max}, RC_{\min}]$ is divided into three partitions each, using *quantiles* [57] so that the influence of noise is lessened. Only strong and positive (negative) interactions are selected. Thereafter, a network connecting the various genes is generated.

### 5.2 The Algorithm

The main steps of the procedure are outlined as follows:

I) Extraction of biclusters by multi-objective genetic algorithm.
II) Determination of pairwise rank correlation between gene pairs.
III) Discretization of the correlation matrix for eliminating the weaker interactions.
IV) Network generation from connectivity matrix (Section 5.3.1)
V) Biological validation (as discussed in Section 5.3.2).

### 5.3 Experimental Results

Data from the budding yeast *S.cerevisiae* is employed for extracting the gene interaction sub-networks.

#### 5.3.1 Network Extraction

*Yeast* cell-cycle CDC28 data [58] is a collection of 6178 genes (attributes) for 17 conditions (time points), taken at 10-minute time intervals covering nearly two cycles. The synchronization of the yeast cell cultures was done using the so-called CDC28 arrest. The experiments were performed using Affymetrix oligonucleotide array. The missing values present in the data set were imputed according to the methodology provided in [59][6]. At first pairwise rank correlation coefficients between gene pairs are computed by eqn. 1 to generate the network architecture from the extracted biclusters. Quantile partitioning is employed next, to choose the strong positive as well as negative correlation links. In this way, the top $\frac{1}{3}$ of the positive and negative links is chosen to be connected in a network. A sample network consisting of three biclusters of sizes 7, 10, and 14, respectively, are shown in Fig. 2. A transcription factor is connected to its target gene by an arrow when such a TF-Target pair is found to exist within any of the biclusters. Gene pairs connected by solid lines depict positive correlation, while those connected by dashed lines are negatively correlated. TFs external to the network, but having targets within the network, are connected to their corresponding targets by dotted arrows. As an example, the TF YHR084W (encircled with solid lines) is a member of the network of 10 genes and has targets in all the three networks. An external TF YJL056C (encircled with dotted lines) has targets in networks of 7 and 10 genes. The biclusters were biologically validated from gene ontology study, based on the statistically significant GO annotation database[7].

### 5.3.2 Biological Validation

During the prediction of regulatory networks [60] the genes YHR084W and YLR351C were reported to form a TF-Target pair. We also obtained the summary of the TF-Target pair YHR084W-YLR351C (Fig. 2) in terms of *Molecular Function, Biological Process and Cellular Component* from the *Saccharomyces Genome Database* (SGD)[8]. From our calculations we have also confirmed that an interaction exists between the target and its TF. It is reported in the database that the biological process involving protein YLR351C is not fully understood as yet and YHR084W has transcription factor activity. It becomes more difficult when one attempts to extract some biologically meaningful information involving these two entities. From such scanty information our method has been able to identify that there exists a link between a TF and its target. From their cellular components we model, as an efficacy of the biclustering, the transcription of YLR351C by YHR084W occurring inside the nucleus, and then the regular translation mechanism follows. In likewise manner for the TF-Target pair of YPL075W and YJR045C (Fig. 2) reported in [59], we obtained their summary from SGD and found YPL075W to be transcriptional activator of genes involved in glycolysis while YJR045C has *ATPase*, enzyme regulator and protein transporter activity. Again we were able to predict that YPL075W is involved in the transcription of YJR045C and would go into the glycolysis process.

One can arrive at similar kind of conclusions, for the rest TF-Target pairs, with a certain definite degree of confidence. As relevant literature in this area are really very sparse a large number negative results is only expected. Our algorithm has not yet detected any false positive or false negative TF-Target pairs, which is consistent with the information available either in the literature or in the databases.

[6] LSimpute: accurate estimation of missing values in microarray data with least squares methods.

[7] http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

[8] A scientific database of the molecular biology and genetics of the yeast *Saccharomyces cerevisiae* - http://db.yeastgenome.org/

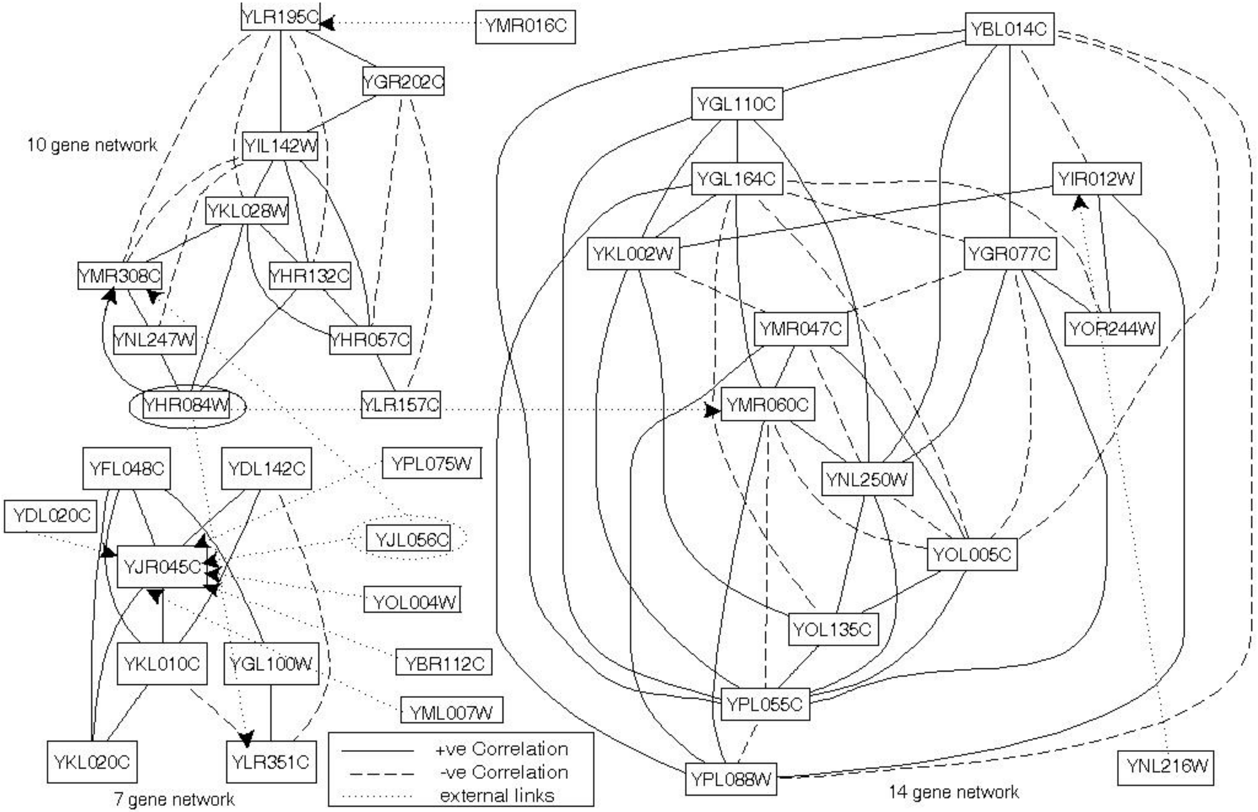


**Fig. 2.** Network (bicluster) of 10 genes connected by transcription factor YHR084W to networks (biclusters) of 7 and 14 genes

## 6 Conclusions and Discussion

In this chapter we have described the extraction of gene interaction networks. This was followed by a soft-computing approach to reverse engineering. Multi-objective evolutionary biclustering selected the co-regulated partitions. Subsequently, rank correlated gene pairs were extracted as a part of the gene interaction subnetworks.

Biologically relevant small biclusters were obtained, using time-series gene expression data from *Yeast*. These were validated using the statistically significant GO annotation database. The pairwise rank correlation coefficients among gene pairs were computed by eqn. 1 followed by the quantile partitioning to select the strong positive as well as negative correlation links. The strongly correlated genes were then chosen to be connected in a network. The TF-Target gene pairs in the network, shown in Fig. 2, were found to exhibit strong correlations. We tried to model the interaction among them from information available in the literature/databases *viz.*, SGD. We have also analyzed the expression profiles of the regulator and the regulated genes, which revealed several complex (time shifted, inverted, and simultaneous, *etc.*) relationships between them. The sparse nature of gene regulatory networks was reflected well on choosing Spearman rank correlation as the similarity measure.

## References

[1] Mitra, S., Pedrycz, W. (eds.): Special Issue on Bioinformatics. Pattern Recognition 39 (2006)

[2] Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of National Academy of Sciences USA 95, 14863–14868 (1998)

[3] Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)

[4] Gasch, A.P., Eisen, M.B.: Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. Genome Biology 3:research 0059.1-0059.22 (2002)

[5] Ji, L., Tan, K.L.: Identifying time-lagged gene clusters using gene expression data. Bioinformatics 21, 509–516 (2005)

[6] Madeira, S.C., Oliveira, A.L.: A Linear Time Biclustering Algorithm for Time Series Gene Expression Data. In: Casadio, R., Myers, G. (eds.) WABI 2005. LNCS (LNBI), vol. 3692, pp. 39–52. Springer, Heidelberg (2005)

[7] Cheng, Y., Church, G.M.: Biclustering of gene expression data. In: Proceedings of ISMB 2000, pp. 93–103 (2000)

[8] Karp, P.D., Ouzounis, C.A., Moore-Kochlacs, C., Goldovsky, L., Kaipa, P., Ahren, D., Tsoka, S., Darzentas, N., Kunin, V., Lopez-Bigas, N.: Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research 19, 6083–6089 (2005)

[9] Keseler, I.M., Collado-Vides, J., Gama-Castro, S., Ingraham, J., Paley, S., Paulsen, I.T., Peralta-Gil, M., Karp, P.D.: EcoCyc: A comprehensive database resource for Escherichia coli. Nucleic Acids Research 33, 334–337 (2005)

[10] Salgado, H., Gama-Castro, S., Peralta-Gil, M., Diaz-Peredo, E., Sanchez-Solano, F., Santos-Zavaleta, A., Martinez-Flores, I., Jimenez-Jacinto, V., Bonavides-Martinez, C., Segura-Salazar, J., Martinez-Antonio, A., Collado-Vides, J.: RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. Nucleic Acids Research 34, D394–D397 (2006)

[11] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology, the gene ontology consortium. Nature Genetics 25, 25–29 (2000)

[12] Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M.: KEGG: Kyoto Encyclopedia of Genes Genomes. Nucleic Acids Research 27, 29–34 (1999)

[13] Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., Barabasi, A.-L.: The large scale organization of metabolic networks. Nature 407, 651–654 (2000)

[14] de Jong, H.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computational Biology 9, 67–103 (2002)

[15] D'haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: From co-expression clustering to reverse engineering. Bioinformatics 16, 707–726 (2000)

[16] Thieffry, D., Huerta, A.M., P´erez-Rueda, E., Collado-Vides, J.: From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in Escherichia coli. BioEssays 20, 433–440 (1998)

[17] Friedman, N., Linial, M., Nachman, I., Pe'er, D.: Using Bayesian networks to analyze expression data. Journal of Computational Biology 7, 601–620 (2000)

[18] Husmeier, D.: Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics 19, 2271–2282 (2003)

[19] Akutsu, T., Miyano, S., Kuhara, S.: Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: Proceedings of Pacific Symposium on Biocomputing, pp. 17–28 (1999)

[20] Liang, S., Somogyi, F.S.: Somogyi Reveal: a general reverse engineering algorithm for inference of genetic network architectures. In: Proceedings of Pacific Symposium on Biocomputing, pp. 18–29 (1998)

[21] Martin, S., Zhang, Z., Martino, A., Faulon, J.-L.: Boolean dynamics of genetic regulatory networks inferred from microarray time series data. Bioinformatics 23, 866–874 (2007)

[22] Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W.: Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18, 261–274 (2002)

[23] Gardner, T.S., di Bernardo, D., Lorenz, D., Collins, J.J.: Inferring genetic networks and identifying compound mode of action via expression profiling. Science 4, 102–105 (2003)

[24] de Jong, H., Page, M.: Search for steady states of piecewise-linear differential equation models of genetic regulatory networks. IEEE Transactions on Computational Biology and Bioinformatics 5, 208–222 (2008)

[25] Zou, M., Conzen, S.D.: A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics 21, 71–79 (2005)

[26] Yu, J., Smith, V.A., Wang, P.P., Hartemink, A.J., Jarvis, E.D.: Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics 20, 3594–3603 (2004)

[27] Kwon, A.T., Hoos, H.H., Ng, R.: Inference of transcriptional regulation relationships from gene expression data. Bioinformatics 19, 905–912 (2003)

[28] Segal, E., Taskar, B., Gasch, A., Friedman, N., Koller, D.: Rich probabilistic models for gene expression. Bioinformatics 17, S243–S252 (2001)

[29] Bar-Joseph, Z., Gerber, G.K., Lee, T.I., Rinaldi, N.J., Yoo, J.Y., Robert, F., Gordon, D.B.: E Fraenkel, Jaahhola TS, Young RA, Gifford DK Computational discovery of gene modules and regulatory networks. Nat. Biotechnol. 21, 1337–1342 (2003)

[30] Cheng, Y., Church, G.M.: Biclustering of gene expression data. In: Proceedings of ISMB 2000, pp. 93–103 (2000)

[31] Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics 1, 24–45 (2004)

[32] Reiss, D.J., Baliga, N.S., Bonneau, R.: Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. BMC Bioinformatics 7, 280 (2006)

[33] Zadeh, L.A.: Fuzzy logic, neural networks, and soft computing. Communications of the ACM 37, 77–84 (1994)

[34] Mitra, S., Acharya, T.: Data Mining: Multimedia, Soft Computing, and Bioinformatics. John Wiley, New York (2003)

[35] Pedrycz, W., Skowron, A., Kreinovich, V.: Handbook of Granular Computing. Wiley, England (2008)

[36] Bargiela, A., Pedrycz, W.: Toward a Theory of Granular Computing for Human-Centered Information Processing. IEEE Transactions on Fuzzy Systems 16, 320–330 (2008)

[37] Takahashi, H., Tomida, S., Kobayashi, T., Honda, H.: Inference of common genetic network using fuzzy adaptive resonance theory associated matrix method. Journal of Bioscience and Bioengineering 96, 154–160 (2003)

[38] Woolf, P.J., Wang, Y.: A fuzzy logic approach to analyzing gene expression data. Physiol Genomics 3, 9–15 (2000)

[39] Du, P., Gong, J., Wurtele, E.S., Dickerson, J.A.: Modeling gene expression networks using fuzzy logic. IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics 35, 1351–1359 (2005)

[40] Kim, S., Dougherty, E.R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J.M., Bittner, M.: Multivariate Measurement of Gene Expression Relationships. Genomics 67, 201–209 (2000)

[41] Huang, J., Shimizu, H., Shioya, S.: Clustering gene expression pattern and extracting relationship in gene network based on artificial neural networks. J Biosci. Bioeng. 96, 421–428 (2003)

[42] Ressom, H., Wang, D., Natarajan, P.: Clustering gene expression data using adaptive double self-organizing map. Physiol. Genomics 14, 35–46 (2003)

[43] Toronen, P., Kolehmainen, M., Wong, G., Castren, E.: Analysis of gene expression data using self-organizing maps. FEBS Lett. 451, 142–146 (1999)

[44] Vohradsky, J.: Neural network model of gene expression. FASEB Journal 15, 846–854 (2001)

[45] Weaver, D.C., Workman, C.T., Stormo, G.D.: Modelling regulatory networks with weight matrices. In: Proceedings of Pacific Symposium on Biocomputing, pp. 112–123 (1999)

[46] Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., Tomita, M.: Dynamic modeling of genetic networks using genetic algorithm and S-system. Bioinformatics 19, 643–650 (2003)

[47] Xiong, M., Li, J., Fang, X.: Identification of genetic networks. Genetics 166, 1037–1052 (2004)

[48] Iba, H., Mimura, A.: Inference of a gene regulatory network by means of interactive evolutionary computing. Information Science 145, 225–236 (2002)

[49] Keedwell, E., Narayanan, A.: Discovering gene networks with a neural-genetic hybrid. IEEE Transactions on Computational Biology and Bioinformatics 2, 231–242 (2005)

[50] Kasabov, N.K.: Knowledge-based neural networks for gene expression data analysis modelling and profile discovery. Biosilico 2, 253–261 (2004)

[51] Cotik, V., Zaliz, R.R., Zwir, I.: A hybrid promoter analysis methodology for prokaryotic genomes. Fuzzy Sets and Systems 152, 83–102 (2005)

[52] Bansal, M., Belcastro, V., Ambesi-Impiombato, A., di Bernardo, D.: How to infer gene networks from expression profiles. Molecular Systems Biology 3, 1–10 (2007)

[53] Zhang, Y., Zha, H., Chu, C.H.: A time-series biclustering algorithm for revealing co-regulated genes. In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2005), pp. 1–6 (2005)

[54] Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition 39, 2464–2477 (2006)

[55] Balasubramaniyan, R., Hllermeier, E., Weskamp, N., Kamper, J.: Clustering of gene expression data using a local shape-based similarity measure. Bioinformatics 21, 1069–1077 (2005)

[56] Das, R., Mitra, S., Banka, H., Mukhopadhyay, S.: Evolutionary biclustering with correlation for gene interaction networks. In: Ghosh, A., De, R.K., Pal, S.K. (eds.) PReMI 2007. LNCS, vol. 4815, pp. 416–424. Springer, Heidelberg (2007)

[57] Davies, G.R., Yoder, D.: Business Statistics. John Wiley & Sons, Inc., London (1937)

[58] Cho, R.J., Campbell, M.J., Winzeler, L.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J., Davis, R.W.: A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell 2, 65–73 (1998)

[59] Bo, T.H., Dysvik, B., Jonassen, I.: Lsimpute: accurate estimation of missing values in microarray data with least squares methods. Nucleic Acids Research 32, 1–8 (2004)

[60] Qian, J., Lin, J., Luscombe, N.M., Yu, H., Gerstein, M.: Prediction of regulatory networks: genome-wide identification of transcription factor targets from gene expression data. Bioinformatics 19, 1917–1926 (2003)

# Uncertain Identification Problems in the Context of Granular Computing

Tadeusz Burczyński[1,2] and Piotr Orantek[1]

[1] Silesian University of Technology, Department for Strength of Materials and Computational Mechanics, Konarskiego 18A, 44-100 Gliwice, Poland
Tadeusz.Burczynski@polsl.pl, Piotr.Orantek@polsl.pl
[2] Cracow Univerity of Technology, Institute of Computer Modelling,
Artificial Intelligence Department, Warszawska 24, 31-155 Cracow, Poland

**Abstract.** The chapter is devoted to applications of selected methods of computational intelligence: evolutionary algorithms and artificial neural networks, in identification of physical systems being under the uncertain conditions. Uncertainties can occur in boundary conditions, in material coefficients or some geometrical parameters of systems and are modeled by three kinds of granularity: interval mathematics, fuzzy sets and theory of probability. In order to evaluate fitness functions the interval, fuzzy and stochastic finite element methods are applied to solve granular boundary-value problems for considered physical systems. Several numerical tests and examples of identification of uncertain parameters are presented.

## 1 Introduction

In the majority engineering cases it is not possible to determine exactly all parameters of the physical systems. It is necessary to introduce some uncertain parameters [2] which describe the granular character of data. Representation of uncertain values may have different forms. It depends of the physical meaning of the considered problem and the assumed model of uncertainty. There are several for-mal frameworks in which information granules can be built [2], among them interval analysis [17], fuzzy sets [25] [12] [13] and random variables [19] can be considered. The aim of an identification problem is to find some unknown parameters of a physical system by minimizing a norm between computed and measured state fields. Due to the fact that some parameters of the physical system have the granular character then the minimized fitness function is also granular.

The evolutionary algorithms [16], as the global optimization technique for searching uncertain values, are applied in finding the interval parameter, fuzzy models, fuzzy controllers, fuzzy rules, random parameters and others [10][11][18][20][21]. In such algorithms, the chromosome consists of uncertain genes. Therefore, the evolutionary operators are modified for uncertain types of data.

This chapter describes a novel idea of uncertain identification problems in the context of granular computing using granular evolutionary algorithms and artificial neural networks. The following kinds of granular models are considered: (i) interval numbers, (ii) fuzzy numbers and (iii) random variables.

The presented idea differs from conventional methods of identification, based on hard computing, in that, unlike hard computing, it is tolerant of imprecision, uncertainty,

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 329–350.
springerlink.com

partial truth, and approximation. In effect, the role model for identification based on soft computing is the human mind.

In the first part of the chapter the uncertain identification problem is formulated. Next, the following optimization algorithms in the granular form are described: (i) an evolutionary algorithm, (ii) a local gradient method supported by artificial neural networks and (iii) a global evolutionary strategy. In the finish part of the chapter several examples of applications of the presented approach for selected granular forms in uncertain identification problems are presented.

## 2 Formulation of Uncertain Identification Problem

Consider a continuous physical system occupying a domain $\Omega \subset E^m (m = 2 \; or \; 3)$ bounded by a boundary $\partial\Omega \equiv \Gamma$ (Fig.1). The system is described by the following boundary value problem:

$$\begin{aligned} &\mathbf{Au} = \mathbf{b}(\mathbf{z}), && \forall \mathbf{z} \in \Omega, \\ &\mathbf{u} = \mathbf{u}_o(\mathbf{z}), && \left.\begin{aligned} &\forall \mathbf{z} \in \partial\Omega_u \equiv \Gamma_u, \\ &\forall \mathbf{z} \in \partial\Omega_p \equiv \Gamma_p, \end{aligned}\right\} \quad \begin{aligned} &\Gamma_u \cup \Gamma_p = \Gamma \\ &\Gamma_u \cap \Gamma_p = \varnothing \end{aligned} \\ &\mathbf{Pu} = \mathbf{p}_o(\mathbf{z}), && \end{aligned} \tag{1}$$

where $\mathbf{A}$ and $\mathbf{P}$ are differential operators which depend on a considered physical problem, $\mathbf{b}$ is a source function, $\mathbf{u}$=$\mathbf{u}(\mathbf{z})$ is a state field which can have various physical meanings, e.g. a displacement and a temperature or an electric potential.

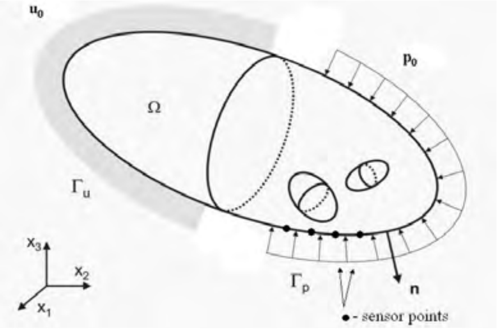


**Fig. 1.** The continuous physical systems described by granular parameters

Some parameters $\mathbf{x} = (x_i), i = 1, 2, .., n,$ of the system as: (i) geometry of the boundary $\Gamma$, (ii) coefficients of operators $\mathbf{A}$ or $\mathbf{P}$ (material coefficients) and (iii) boundary conditions are modeled by uncertain (granular) values [17][25][13][19]. The aim of the identification problem is to find the unknown vector $\mathbf{x}$ of parameters of the system having measured granular information about some state fields.

From the mathematical point of view, the identification problem [4][5] is expressed as minimization of a special objective function $f$ with respect to unknown granular parameters $\mathbf{x}$ :

$$\min_{\mathbf{x}} f, \quad f \equiv \sum_{i=1}^{m} \int_{\Gamma} \left[\mathbf{u}(\mathbf{z}) - \hat{\mathbf{u}}(\mathbf{z})\right]^2 \delta\left(\mathbf{z} - \mathbf{z}^i\right) d\Gamma(\mathbf{z}) \tag{2}$$

where: $\mathbf{u}(\mathbf{z})$ - a computed field of state variables (e.g. displacements), $\hat{\mathbf{u}}(\mathbf{z})$ - a measured field of state variables, $\mathbf{z} = \mathbf{z}^i$ - $i$-th sensor point, $m$ – a number of sensor points, $\delta$ - the Dirac function.

# 3 Granular Evolutionary Algorithm

Minimization of the granular function $f$ (2) with respect to unknown granular parameters is performed by the granular evolutionary algorithm with granular operators and granular representation of the data. The chromosomes contain granular genes which play the role of identification parameters. Each gene decides about the heredity of one or a few characteristics. The individuals can be modified by means of the granular operators. The evolutionary operators generate new chro-mosomes. The next step is the operator of the selection. It creates a new genera-tion, which contains better chromosomes. All steps are repeated until the stop condition is fulfilled.

In the granular evolutionary algorithm an individual expresses a granular solution. In each generation the granular evolutionary algorithm contains a population of solutions. Each solution is evaluated, and as the result a granular value of the fitness function is obtained.

## 3.1 The Granular Representation of Chromosomes

In most cases genes are represent by the real values. The granular evolutionary algorithm works on the granular data, so the gene should be modified to granular data. The following cases are considered: (i) interval genes, (ii) fuzzy genes and (iii) random genes.

### 3.1.1 The Interval Chromosome

In the interval case the gen $[x] = [\underline{x}, \overline{x}]$ is described by the central value $cv([x]) = (\underline{x} + \overline{x})/2$ and the radius $r([x]) = (\underline{x} - \overline{x})/2$.

Therefore the interval chromosome expressed by:

$$\left[[x_1],[x_2],...,[x_i],...,[x_n]\right] \tag{3}$$

can be replaced by the real-coded chromosome:

$$\left[(cv_1,r_1),(cv_2,r_2),...,(cv_i,r_i),...,(cv_n,r_n)\right] \tag{4}$$

where: $[x_i] = (cv_i, r_i)$.

### 3.1.2 The Fuzzy Chromosome

In the fuzzy case the gene $x$ can be considered as a fuzzy set. The fuzzy set is considered as a set of pairs of the $x$ and the membership function $\mu(x)$. When the fuzzy set is convex and normal and the membership function is continuous, the fuzzy set is the fuzzy number. The concept of alpha-cuts plays the important role in the theory of fuzzy sets.

An alpha-cut of a fuzzy number $x$ is an interval that contains all the numbers of $x$ that have the membership value of $x$ greater than or equal to alpha. In this case the fuzzy number can be replaced by a set of the interval values, which are stretched on the adequate levels (alpha-cuts) of the fuzzy value. This approach has some advantages. For each alpha-cut the interval arithmetic operators can be used. It is possible to obtain different forms of the fuzzy values due to the generation of a few alpha-cuts and corresponding them interval values $\left[\underline{x},\overline{x}\right]$. The forms can be symmetric or not symmetric. They describe some characteristic forms of the fuzzy values, and permit to build a new form of the fuzzy value, too. Finally, each gene $x$ is expressed as the real value: the *central value* $cv(x)$ and a set of parameters $a^i(x)$ and $b^i(x)$, ($i$=1,…,$M$, where $M$ is a number of alpha-cuts) which define distances between $cv(x)$ and edges of intervals. It is possible to introduce some constraints on the $cv(x)$ and non-symmetric constraints on widths of intervals using the parameters $a^i(x)$ and $b^i(x)$.

Therefore, the fuzzy chromosome expressed by:

$$\mathbf{x}=\left[x_1,x_2,...,x_i,...,x_n\right] \tag{5}$$

can be replaced by the real-coded chromosome (for $M$=2):

$$\left[\left(a_1^1,a_1^2,cv_1,b_1^2,b_1^1\right),...,\left(a_i^1,a_i^2,cv_i,b_i^2,b_i^1\right),...,\left(a_n^1,a_n^2,cv_n,b_n^2,b_n^1\right)\right] \tag{6}$$

where: $x_i=\left(a_i^1,a_i^2,cv_i,b_i^2,b_i^1\right)$.

### 3.1.3 The Stochastic Chromosome

In the theoretical model of random phenomena the basic role is played by the probability space ($G$, $\mathcal{F}$, $P$). The set $G$, called the *space of elementary events* represents all the possible simplest outcomes of a trial associated with a given random phenomenon. $\mathcal{F}$ is a $\sigma$- algebra of subset of $G$. The elements of the $\mathcal{F}$ are called *random events*. $P$ is a *probability* defined on $\mathcal{F}$ [19].

In the random case the gene is represented by a random variable, which is a real function $X_i=X_i(\gamma)$, $\gamma\in G$, defined on a sample space $G$ and measurable with respect to $P$: i.e., for every real number $x_i$, the set $\{\gamma: X_i(\gamma)<x_i\}$ is an event in $\mathcal{F}$. The chromosome $\mathbf{X}(\gamma)$ is a function (measurable respect to $P$) which takes every element $\gamma\in G$ into a point $\mathbf{x}\in R^n$ [19].

The chromosome is expressed as a random vector:

$$\mathbf{X}(\gamma)=[X_1(\gamma), X_2(\gamma),..., X_i(\gamma),..., X_n(\gamma)] \tag{7}$$

which has an $n$-dimensional Gaussian distribution of the probability density function, given as follows:

$$p(x_1,x_2,...,x_i,...,x_n)=\frac{1}{(2\pi)^{n/2}\sqrt{|\mathbf{K}|}}\left[-\frac{1}{|\mathbf{K}|}\sum_{i,j=1}^{n}|K_{ij}|(x_i-m_i)(x_j-m_j)\right] \tag{8}$$

where $|\mathbf{K}|\neq 0$ is the determinant of the matrix covariances, $\mathbf{K}=\left[k_{ij}\right]$, *i, j* = 1,2,…,*n*, where $k_{ij}=\mathbf{E}\left[\left(X_i-m_i\right)\left(X_j-m_j\right)\right]$, $|K_{ij}|$ is the co-factor of the element $k_{ij}$ the matrix $\mathbf{K}$ and $m_i=\mathbf{E}\left[X_i\left(\gamma\right)\right]$ is the mean value of $X_i\left(\gamma\right)$.

It is assumed that random genes are independent random variables. The joint probability density function is expressed by the probability density functions of single random genes as follows:

$$p(x_1,x_2,...,x_i,...,x_n)=p_1(x_1)p_2(x_2)....p_i(x_i)....p_n(x_n) \tag{9}$$

where:

$$p_i\left(x_i\right)=N\left(m_i,\sigma_i\right)=\frac{1}{\sigma_i\sqrt{2\pi}}\exp\left[-\frac{\left(x_i-m_i\right)^2}{2\sigma_i^2}\right] \tag{10}$$

is the probability density function of the random gene $X_i(\gamma)$, where $\sigma_i$ denotes the standard deviation of $X_i(\gamma)$.

It can be seen that if the random genes $X_i(\gamma)$, i=1,2,…,*n*, are random independent Gaussian variables, two moments describe the probability density function of the random variable $X_i(\gamma)$.

The stochastic chromosome (7) is replaced by a vector:

$$\left[\left(m_1,\sigma_1\right),\left(m_2,\sigma_2\right),...,\left(m_i,\sigma_i\right),...,\left(m_n,\sigma_n\right)\right] \tag{11}$$

which is described by moments $m_i$ and $\sigma_i$.

## 3.2 The Granular Operators

### 3.2.1 The Granular Mutation

In the interval case two types of the mutation operators are applied. In both cases the modified gene $x_j$ is randomly selected from the chromosome **x**=$[x_1,x_2,...,x_j,...,x_n]$.

In the first type of the mutation (mutation I) the central value $cv(x_j)$ of the *j*-th interval value $x_j$ is modified. The operator is expressed by the following equation:

$$h(x_j^*)=h(x_j)+H_h \tag{12}$$

where: *h=cv* for each gene, $H_h$ – random value (with Gaussian distribution), *j*=1..*n* is the number of the gene.

The second type of the mutation operators (mutation II) concentrates/deconcentrates the interval value. The mutation changes the radius $r(x_j)$ according to the equation (12), where $h=r$ for each gene. Therefore, two types of the mutation operator are introduced, both can work together or independently.

In the fuzzy case two types of the mutation operators are also applied. In both cases the modified gene $x_j$ is randomly selected from the chromosome $\mathbf{x}=[x_1,x_2,...,x_j,...,x_n]$.

In the first type of the mutation (mutation I) the central value $cv(x_j)$ of the *j*-th fuzzy value $x_j$ is modified. The operator is expressed by the equation (12), where $h=cv$.

The second type of the mutation operators (mutation II) concentrates/deconcentrates the fuzzy value $x_j$. The mutation changes the distances $a^i(x_j)$ or $b^i(x_j)$ by equation (12) where: $h= a_i, b_i$.

This operator is considered as symmetric ( $a^i(x_j)$ and $b^i(x_j)$ are changed by means of the same value), and non-symmetric ones. The operator can change only the selected alpha-cut. Therefore, two types of the mutation operator is introduced, both can work together or independently.

In the random variables case, two types of the mutation operators are also applied. In both cases the modified gene $x_j$ is randomly selected from the chromosome $\mathbf{x}=[x_1,x_2,...,x_j,...,x_n]$. The first type changes the first normal moment $m$ using formula (12), where: $h=m$. The second type changes the standard deviation of the *j*-th random variable using formula (12), where: $h=\sigma$ [18].

### 3.2.2 The Granular Crossover

The granular arithmetic crossover operator is proposed in the granular evolutionary algorithm. The crossover creates two offspring individuals $\mathbf{x}^* = \left[x_1^*, x_2^*,...,x_j^*,...,x_n^*\right]$ and $\mathbf{y}^* = \left[y_1^*, y_2^*,...,y_j^*,...,y_n^*\right]$ on the basis of two parent chromosomes $\mathbf{x} = \left[x_1, x_2,...,x_j,...,x_n\right]$ and $\mathbf{y} = \left[y_1, y_2,...,y_j,...,y_n\right]$.

The selected parameters of the *j*-th genes of the offspring chromosomes are expressed by the following equations (interval cases):

$$h(x_j) = \lambda h(x_j) + (1-\lambda)h(y_j) \tag{13}$$

$$h(y_j^*) = \lambda h(y_j^*) + (1-\lambda)h(x_j^*) \tag{14}$$

where: $h=cv, r$, and $\lambda \in [0,1]$ is a random value with the uniform distribution.

In the fuzzy case the selected parameters of the *j*-th genes of the offspring chromosomes are expressed by the equations (13) and (14), where $h=cv,a^i,b^i$.

In the random variables case the offspring chromosomes are expressed by equations (13) and (14), where $h=m,\sigma$. More details are shown in [18].

### 3.2.3 The Granular Selection

The last modified operator for the interval, fuzzy values and random variables is the selection operator. This operator is constructed on the basis of the well known tournament selection. In this selection the fitness function values *f* are compared, and the better chromosome wins more often. Therefore the special strategy of comparison of two granular values $f_1$ and $f_2$ is proposed.

In interval case the special conditions are constructed:

$$h_{f_1} < h_{f_2} \tag{15}$$

where: $h=cv,r$.

In the fuzzy case the condition (15) is checked, where: $h=cv,\ a^i,\ b^i$ . In the stochastic case the condition (15) is checked, where: $h=m,\ \sigma$ [18].

### 3.3 The Granular Fitness Function

One of the most important steps of the evolutionary algorithm is the evaluation of the fitness function. If the design variables are deterministic, the fitness function result is also deterministic. In the case of solving the granular optimization problems, the problem of evaluating the fitness function is much more complicated. A few ways to estimate the results are possible in this case. In the case of simple mathematical functions the basic arithmetic operators {+;-;*;/} for granular representation are used.

Unfortunately, in many physical problems the fitness function can be examinated after solving the interval/fuzzy/stochastic boundary-value problem. The boundary-value problems can be solved by means of the interval/fuzzy/stochastic boundary element method or the interval/fuzzy/stochastic finite element method [8][9][14].

### 3.4 Testing the Granular Evolutionary Algorithm

The aim of the test is to find the granular vector **x** which minimizes the function:

$$f = f(\mathbf{x}) = \sum_{i=1}^{n}\left[\frac{1}{\pi}\left|x_i - 0.7\right|^{\frac{\pi}{3}}\left(\frac{\pi}{2} - \cos\left(2\pi p\left|x_i - 0.7\right|^{\frac{\pi}{3}}\right)\right)\right] \tag{16}$$

where: $n$ – the number of granular design decision variables $x_i$ , $p$ – the number of the optimum.

In the first step of examination, the best (optimal) probabilities of the mutation (*pm*) and the crossover (*pc*) operators were searched. In the second stage the best population size (*ps*) was searched. For each combination (*n*=1..5, *p*=1..5) the 10000 independent experiments were run. The optimal probabilities and population size of granular evolutionary algorithms (interval, fuzzy and stochastic) are included in Tables 1 ÷ 4.

**Table 1.** The optimal probabilities *pm*, *pc* and population size *ps* of the granular evolutionary algorithm (interval case)

| *n* | *p* | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
| | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* |
| 1 | 0.4 | 0.2 | 4 | 0.4 | 0.2 | 3 | 0.3 | 0.1 | 3 | 0.3 | 0.1 | 4 | 0.3 | 0.1 | 3 |
| 2 | 0.4 | 0.1 | 4 | 0.4 | 0.1 | 4 | 0.4 | 0.1 | 4 | 0.3 | 0.1 | 4 | 0.3 | 0.1 | 4 |
| 3 | 0.3 | 0.1 | 4 | 0.3 | 0.1 | 5 | 0.3 | 0.1 | 4 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 5 |
| 4 | 0.3 | 0.1 | 5 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 4 |
| 5 | 0.3 | 0.1 | 5 | 0.2 | 0.1 | 5 | 0.2 | 0.1 | 4 | 0.3 | 0.1 | 4 | 0.2 | 0.1 | 5 |

**Table 2.** The optimal probabilities *pm*, *pc* and population size *ps* of the granular evolutionary algorithm (fuzzy case, 2 alpha cuts)

| *n* | *p* | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
| | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* |
| 1 | 0.4 | 0.1 | 5 | 0.4 | 0.1 | 4 | 0.3 | 0.1 | 5 | 0.3 | 0.1 | 5 | 0.3 | 0.1 | 6 |
| 2 | 0.4 | 0.1 | 4 | 0.4 | 0.1 | 5 | 0.4 | 0.1 | 5 | 0.4 | 0.1 | 7 | 0.4 | 0.1 | 8 |
| 3 | 0.5 | 0.2 | 11 | 0.5 | 0.2 | 15 | 0.5 | 0.2 | 15 | 0.5 | 0.2 | 18 | 0.5 | 0.2 | 19 |
| 4 | 0.5 | 0.2 | 10 | 0.5 | 0.2 | 15 | 0.5 | 0.2 | 18 | 0.5 | 0.2 | 18 | 0.5 | 0.2 | 20 |
| 5 | 0.5 | 0.2 | 11 | 0.5 | 0.2 | 16 | 0.5 | 0.2 | 19 | 0.5 | 0.2 | 21 | 0.5 | 0.2 | 23 |

**Table 3.** The optimal probabilities *pm*, *pc* and population size *ps* of the granular evolutionary algorithm (fuzzy case, 3 alpha cuts)

| *n* | *p* | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
| | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* |
| 1 | 0.4 | 0.1 | 4 | 0.3 | 0.1 | 6 | 0.3 | 0.1 | 8 | 0.3 | 0.1 | 9 | 0.3 | 0.1 | 11 |
| 2 | 0.4 | 0.1 | 4 | 0.4 | 0.1 | 6 | 0.4 | 0.1 | 6 | 0.4 | 0.1 | 9 | 0.4 | 0.1 | 11 |
| 3 | 0.4 | 0.2 | 8 | 0.4 | 0.2 | 14 | 0.4 | 0.2 | 13 | 0.4 | 0.2 | 13 | 0.4 | 0.2 | 15 |
| 4 | 0.5 | 0.2 | 10 | 0.5 | 0.2 | 19 | 0.4 | 0.2 | 17 | 0.4 | 0.2 | 21 | 0.5 | 0.2 | 22 |
| 5 | 0.5 | 0.2 | 11 | 0.5 | 0.2 | 16 | 0.5 | 0.2 | 19 | 0.5 | 0.2 | 24 | 0.5 | 0.2 | 27 |

**Table 4.** The optimal probabilities *pm*, *pc* and population size *ps* of the granular evolutionary algorithm (stochastic case)

| *n* | *p* | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
| | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* | *pm* | *pc* | *ps* |
| 1 | 0.4 | 0.1 | 4 | 0.4 | 0.1 | 5 | 0.3 | 0.1 | 4 | 0.3 | 0.1 | 9 | 0.3 | 0.1 | 11 |
| 2 | 0.3 | 0.1 | 4 | 0.3 | 0.1 | 6 | 0.3 | 0.1 | 9 | 0.4 | 0.1 | 9 | 0.4 | 0.1 | 11 |
| 3 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 5 | 0.2 | 0.1 | 5 | 0.4 | 0.2 | 13 | 0.4 | 0.2 | 15 |
| 4 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 5 | 0.2 | 0.1 | 5 | 0.4 | 0.2 | 21 | 0.5 | 0.2 | 22 |
| 5 | 0.2 | 0.1 | 4 | 0.2 | 0.1 | 5 | 0.2 | 0.1 | 4 | 0.5 | 0.2 | 24 | 0.5 | 0.2 | 27 |

The obtained optimal parameters of the mutation, the crossover and the population size are very similar to the optimal parameters in real-coded cases, which were tested in earlier examinations. The probability of the mutation varies mainly between 0.3 and 0.4. The probability of the crossover is close to the value 0.4 in many cases. The population size is relatively small and depends on the number of variables. Therefore all parameters are very easy in setting and the expected computation time should be relatively small.

# 4 The Local Gradient Method Supported by Artificial Neural Networks

## 4.1 The (RBF) Artificial Neural Networks

The approximation problem is one of the well known applications of artificial neural networks [1].

Consider artificial neural networks with radial (RBF) active functions. The number of neurons in the input layer is equal to the number of design variables of the approximated fitness function. In the output layer there is only one neuron, its output value plays the role of the fitness function.

The number of radial basis functions depends on the degree of difficulty of the function. The output value of neurons in the hidden layer in the case of the network (RBF) is expressed by:

$$e_{1i} = f\left(u_i\right) = e^{-0.5u_i} \tag{17}$$

where:

$$u_i = \sum_{n=1}^{N} \left( \frac{e_{1k} - t_k^i}{\sigma_k^i} \right)^2 \tag{18}$$

$t_i$ – a center of $i$-th radial function, $\sigma_i$ – a standard deviation of $i$-th radial function.

The output value of the whole network (RBF) is given as follows:

$$e_{21} = \sum_{i=1}^{N} W_i e_{1i} + W_0 \tag{19}$$

where: $W_i$ and $W_0$ - weights. Detailed description can be found in [15].

## 4.2 The Sensitivity of the Neural Network

The sensitivity of the output signal $e_{21}$ of the (RBF) network in the some $z$ input value $e_{0z}$ is expressed by:

$$\frac{de_{21}}{de_{0z}} = \sum_{n_1=1}^{I_1} \frac{ds_{1n_1}}{de_{0z}} \cdot \frac{de_{1n_1}}{ds_{1n_1}} \cdot \frac{ds_{2n_1}}{de_{1n_1}} \cdot \frac{de_{21}}{ds_{21}} \tag{20}$$

The formula (20) can be expressed by the simpler form [18]:

$$\frac{de_{21}}{de_{0z}} = \sum_{n_1=1}^{I_1} \left( -w^{n_1} \exp\left( -\frac{1}{2} u^{n_1} \right) \frac{x_{0z} - t_{0z}^{n_1}}{\left(\sigma_{0z}^{n_1}\right)^2} \right) \tag{21}$$

This way of function sensitivity computation was tested [18] and gives positive and promised results.

### 4.3 The Local Gradient Method

The proposed local optimization method [18] is a combination of the classical gradient method (the steepest descent method) and the (RBF) artificial neural network.

At the beginning of the method a set (cloud) of fuzzy points in the function domain is generated. In order to perform the optimization process the network is constructed. The number of neurons in the input layer is equal to the number of design variables of the approximated fitness function. In the output layer there is only one neuron, its output value plays the role of the fitness function.

In each iteration of the presented optimization method a few steps are performed.

In the first step the set of training vectors of the network is created. In the first iteration the set is created on the basis of the cloud of points. The coordinates of points play the role of the input values of the network, the fitness values in points play the role of output value of the network.

In the second step the network is trained. In the next, third step, the optimization process is carried out. The steepest descent method of optimization is used. The network as the fitness function approximation is used. The gradient formula (21) is employed in computation.

For a point, which is a result of optimization (found in step 3), the actual fitness function is computed.

In the last step the stop condition is checked. In the case, in which the condition is true, the point is treated as the result of the optimization process. If this condition is false, this point is added to the training vector set and the next iteration is carried out (go to step 1).

### 4.4 The Granular Local Gradient Method

Presented in the previous subsection the local gradient method is based in the real (non granular) representation. It can be extended to different models of granular cases.

A special multilevel artificial neural network is used as the approximation tool of the granular problem [18].

In the interval case the multilevel artificial neural network has 2 levels. The first level corresponds with the central value of the interval and the second one with the radius of an interval.

In fuzzy case the number of levels is equal to 1+2*cm*, where *cm* is the number of alpha-cuts. Each level corresponds with a selected parameter of the fuzzy number (the *central value cv* and a set of parameters $a^i$ and $b^i$, ($i$=1,…,$M$, where $M$ is a number of alpha-cuts)).

In stochastic case the number of levels is equal to the number of moments, which describes the random number. In presented works the random number is described by two moments ($m$ - the mean value and $\sigma$ - standard deviation) therefore the number of level is equal to 2.

This approach and its application results are described more detailed in [18].

## 5 The Two-Stage Granular Evolutionary Strategy

The main idea of creating the two-stage strategy [15] is the coupling of the advantages of evolutionary and gradient optimization methods aided by neuro-computing.

The evolutionary algorithms are very useful in finding the global optimum, but it is very time consuming. The gradient methods can find the optimum precisely, but they need information about sensitivity of the objective function.

The strategy in the first stage uses some properties of the evolutionary algorithms.

They are procedures which search the optimum in the feasible space of solutions. The evolutionary algorithm generates a cluster of points. The cluster is positioned closely to the optimum and there is a great possibility that this optimum is global. Sometime there is a risk that points are located close to the more than one optimum. In this case the second stage (a local method) can work unstably. It can be solved in a few ways. One is to introduce a parameter which describes the maximum size of the cluster. The parameter can be expressed by a radius of a region in the domain. The center of the region is equal to the best solution of the evolutionary algorithm. All points which are inside the region, belong to the cloud of points. This approach is characterized by a variable number of training vectors. In this case an alternative parameter is introduced. The parameter defines the maximum number of points in the cloud.

In the second stage of the strategy several best points in this region are selected. Then, these points play the role of the cloud and as previously shown, the local method is performed. This method is based on the gradient method, but the sensitivity analysis is evaluated by the neuro-computing.

The described strategy combines advantages of the previous described methods, and avoids disadvantages.

The crucial problem is the moment of transition from the first stage to the second one. Some experience allows taking parameters of the strategy, for which the strategy can find the optimum earlier than the evolutionary algorithm, which is used separately.

In the general case the moment of transition can depend on some parameters of the first stage: (i) the changes of the fitness function of the best chromosome, (ii) the size of the clusters of chromosomes, (iii) the diversification of the population and many others. Detailed description can be found in [18].

## 6 Examples of Application in Uncertain in Identification Problems

In all examples presented below one assumes that measured state fields $\hat{\mathbf{u}}$ are simulated numerically by solutions of the granular boundary-value problem (1) for actual parameters.

### 6.1 The Identification of the Boundary Conditions in 3-D Structures

Consider a 3-D elastic system discretized by the cubic finite elements [7]. The dimensions and loading of this system is presented in Figure 2. There are four different kinds of loading $F_1$, $F_2$, $F_3$ and $F_4$, which are imposed on the four different parts of 3-D structure. Moreover this structure is fixing in the bottom part (Figure 2). Due to the symmetry of the structure only two unknown forces $F_1$ and $F_2$ are identified on the basis of displacements measured in selected sensor points.

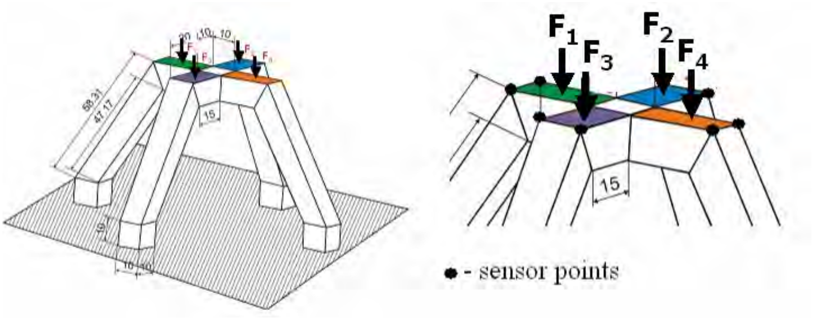


**Fig. 2.** The mechanical structure described by granular parameters

The actual and found values of F$_1$ and F$_2$ are presented in the Table 5 (interval case), Table 6 (fuzzy case) and Table 7 (stochastic case).

The following parameters of the granular strategy in the interval and stochastic cases were assumed: a population size: 50, a number of generations: 50, a probability of the mutation: 0.2, a probability of the crossover 0,1. The following parameters of strategy in the fuzzy case were assumed: a population size: 100, a number of generations: 75, a probability of the mutation: 0.2, a probability of the crossover 0,1.

In this example of identification of loads imposing on the elastic system the evolutionary algorithm correctly found a region of the global optimum in all cases of data granularity. The local neuro-gradient method found the optimum with the high accuracy.

**Table 5.** The results of interval identification

| | $\underline{F}_1$ | $\overline{F}_1$ | $\underline{F}_2$ | $\overline{F}_2$ |
|---|---|---|---|---|
| Min | -5.00 | -5.00 | -5.00 | -5.00 |
| Max | 205.00 | 205.00 | 205.00 | 205.00 |
| Actual | 48.00 | 51.00 | 149.00 | 152.00 |
| After 1$^{st}$ stage | 47.55 | 52.45 | 148.57 | 153.99 |
| After 2$^{nd}$ stage | 48.00 | 51.00 | 149.00 | 152.00 |

**Table 6.** The results of fuzzy identification (first and second $\alpha$-cuts)

| alpha cut | first | | | | second | | | |
|---|---|---|---|---|---|---|---|---|
| | $\underline{F}_1$ | $\overline{F}_1$ | $\underline{F}_2$ | $\overline{F}_2$ | $\underline{F}_1$ | $\overline{F}_1$ | $\underline{F}_2$ | $\overline{F}_2$ |
| Min | -5.00 | -5.00 | -5.00 | -5.00 | -5.00 | -5.00 | -5.00 | -5.00 |
| Max | 205.00 | 205.00 | 205.00 | 205.00 | 195.00 | 205.00 | 195.00 | 205.00 |
| Actual | 48.00 | 51.00 | 149.00 | 152.00 | 49.00 | 51.00 | 150.00 | 151.00 |
| After 1$^{st}$ stage | 47.46 | 53.21 | 148.46 | 154.32 | 48.77 | 51.53 | 149.95 | 151.23 |
| After 2$^{nd}$ stage | 48.00 | 51.00 | 149.00 | 152.00 | 49.00 | 51.00 | 150.00 | 151.00 |

**Table 7.** The results of stochastic identification

| | $F_1$ | | $F_2$ | |
|---|---|---|---|---|
| | m | σ | m | σ |
| Min | 0.00 | 0.00 | 0.00 | 0.00 |
| Max | 200.00 | 5.00 | 200.00 | 5.00 |
| Actual | 50.00 | 1.00 | 150.00 | 2.00 |
| After 1st stage | 48.67 | 0.54 | 151.92 | 2.14 |
| After 2nd stage | 50.00 | 1.00 | 150.00 | 2.00 |

## 6.2 The Identification of Boundary Conditions in Elastoplastic Structures

A mechanical system presented in Figure 3 is considered [7]. The system is modeled as an elastoplastic body, fixed on the left side in the hole and loaded using two fields of tractions $p_1$ and $p_2$. Equivalent stresses values were greater than the yield stress for the used material. The tests were performed using 11 sensor points. The position of sensor points is presented in Figure 3.

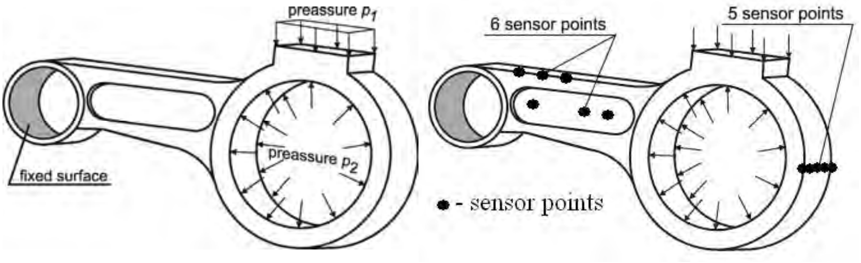


**Fig. 3.** The elastoplastic structure with boundary conditions and sensor points

The actual and found values of tractions p1 and p2 are presented in the Table 8 (interval case), Table 9 (fuzzy case) and Table 10 (stochastic case).

The following parameters of the granular strategy in the interval and stochastic cases were assumed: a population size: 50, a number of generations: 100, a probability of the mutation: 0.2, a probability of the crossover 0.1. The following parameters of the

**Table 8.** The results of interval identification

| | $\underline{p}_1$ | $\overline{p}_1$ | $\underline{p}_2$ | $\overline{p}_2$ |
|---|---|---|---|---|
| Min | -0.90 | -0.90 | -0.90 | -0.90 |
| Max | 11.00 | 11.00 | 11.00 | 11.00 |
| Actual | 1.80 | 2.10 | 7.90 | 8.20 |
| After 1st stage | 1.77 | 2.33 | 7.88 | 8.03 |
| After 2nd stage | 1.80 | 2.10 | 7.90 | 8.20 |

**Table 9.** The results of fuzzy identification (first and second alpha-cuts)

| alpha-cut | first | | | | second | | | |
|---|---|---|---|---|---|---|---|---|
| | $\underline{p}_1$ | $\overline{p}_1$ | $\underline{p}_2$ | $\overline{p}_2$ | $\underline{p}_1$ | $\overline{p}_1$ | $\underline{p}_2$ | $\overline{p}_2$ |
| Min | -0.90 | -0.90 | -0.90 | -0.90 | -0.90 | -0.90 | -0.90 | -0.90 |
| Max | 11.00 | 11.00 | 11.00 | 11.00 | 9.00 | 11.00 | 9.00 | 11.00 |
| Actual | 1.80 | 2.10 | 7.90 | 8.20 | 1.90 | 2.10 | 8.00 | 8.10 |
| After 1st stage | 1.68 | 2.45 | 7.77 | 8.16 | 1.95 | 2.23 | 7.83 | 8.09 |
| After 2nd stage | 1.80 | 2.10 | 7.90 | 8.20 | 1.90 | 2.10 | 8.00 | 8.10 |

**Table 10.** The results of stochastic identification

| | $p_1$ | | $p_2$ | |
|---|---|---|---|---|
| | m | σ | m | σ |
| Min | 0.00 | 0.00 | 0.00 | 0.00 |
| Max | 10.00 | 0.50 | 10.00 | 0.50 |
| Actual | 2.00 | 0.10 | 8.00 | 0.20 |
| After 1st stage | 1.96 | 0.02 | 7.83 | 0.32 |
| After 2nd stage | 2.00 | 0.10 | 8.00 | 0.20 |

granular strategy in the fuzzy case were assumed: a population size: 50, a number of generations: 50, a probability of the mutation: 0.2, a probability of the crossover 0.1.

The aim of the example was to find traction fields in the elastoplastic system. Due to the application of the two stage strategy the optimum was found with the high accuracy and time of identification was short.

### 6.3 The Identification of Beams Distribution

Consider a rectangular plate with the hole, reinforced by beams of circular cross-sections as shown in Figure 4.

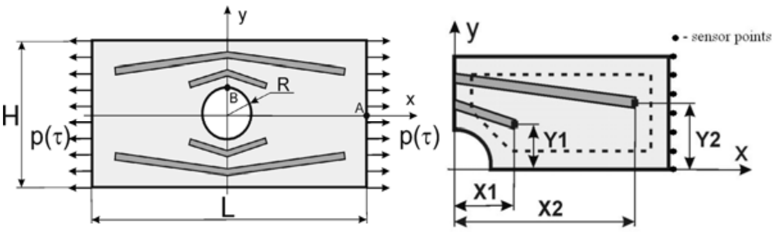


**Fig. 4.** Reinforced plate (modelled as the plate and beams), design variables and constraints

The plate is stretched by the uniformly distributed load applied at the left and right edge [7]. The value of the load is p=10 MPa. The length and the height of the structure and the hole radius are L=10 cm, H=5 cm and R=1 cm, respectively. The thickness of the plate is g=1 cm, the diameter of each beam is d=0.3 cm.

The aim of the identification problem is to find coordinates of ends of beams (design variables $X_1$,$X_2$,$Y_1$ and $Y_2$ shown in Figure 4).

It is assumed that the plate is symmetrical with respect to two axes thus only the quarter of the structure (the upper right part) with two beams and the appropriate boundary conditions at axes are modeled.

The sensor points are located on the external boundary.

The actual and found values of unknown parameters $X_1$,$X_2$,$Y_1$ and $Y_2$ are presented in the Table 11 (interval case), Tables 12 and 13 (fuzzy case) and Table 14 (stochastic case).

The following parameters of the granular strategy in the interval and stochastic cases were assumed: a population size: 50, a number of generations: 50, a probability of the mutation: 0.2, a probability of the crossover 0.1. The following parameters of the

**Table 11.** The results of interval identification

| | $\underline{X}_1$ | $\overline{X}_1$ | $\underline{Y}_1$ | $\overline{Y}_1$ | $\underline{X}_2$ | $\overline{X}_2$ | $\underline{Y}_2$ | $\overline{Y}_2$ |
|---|---|---|---|---|---|---|---|---|
| Min | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 |
| Max | 3.70 | 3.70 | 2.20 | 2.20 | 3.70 | 3.70 | 2.20 | 2.20 |
| Actual | 1.45 | 1.55 | 1.15 | 1.25 | 2.95 | 3.05 | 1.95 | 2.05 |
| After 1st stage | 1.41 | 1.73 | 1.10 | 1.20 | 3.00 | 4.20 | 1.90 | 2.17 |
| After 2nd stage | 1.45 | 1.55 | 1.15 | 1.25 | 2.95 | 3.05 | 1.95 | 2.05 |

**Table 12.** The results of fuzzy identification (first alpha-cut)

| | $\underline{X}_1$ | $\overline{X}_1$ | $\underline{Y}_1$ | $\overline{Y}_1$ | $\underline{X}_2$ | $\overline{X}_2$ | $\underline{Y}_2$ | $\overline{Y}_2$ |
|---|---|---|---|---|---|---|---|---|
| Min | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 |
| Max | 3.70 | 3.70 | 2.20 | 2.20 | 3.70 | 3.70 | 2.20 | 2.20 |
| Actual | 1.45 | 1.55 | 1.15 | 1.25 | 2.95 | 3.05 | 1.95 | 2.05 |
| After 1st stage | 1.38 | 2.92 | 1.20 | 1.49 | 2.98 | 3.05 | 1.95 | 2.04 |
| After 2nd stage | 1.45 | 1.55 | 1.15 | 1.25 | 2.95 | 3.05 | 1.95 | 2.05 |

**Table 13.** The results of fuzzy identification (second alpha-cut)

| | $\underline{X}_1$ | $\overline{X}_1$ | $\underline{Y}_1$ | $\overline{Y}_1$ | $\underline{X}_2$ | $\overline{X}_2$ | $\underline{Y}_2$ | $\overline{Y}_2$ |
|---|---|---|---|---|---|---|---|---|
| Min | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 | 0.80 |
| Max | 3.30 | 3.70 | 1.80 | 2.20 | 3.30 | 3.70 | 1.80 | 2.20 |
| Actual | 1.49 | 1.52 | 1.18 | 1.23 | 3.00 | 3.05 | 2.00 | 2.02 |
| After 1st stage | 1.43 | 1.53 | 1.20 | 1.43 | 2.99 | 3.02 | 1.98 | 2.02 |
| After 2nd stage | 1.49 | 1.52 | 1.18 | 1.23 | 3.00 | 3.05 | 2.00 | 2.02 |

**Table 14.** The results of stochastic identification

| | $X_1$ | | $Y_1$ | | $X_2$ | | $Y_2$ | |
|---|---|---|---|---|---|---|---|---|
| | m | σ | m | σ | m | σ | m | σ |
| Min | 0.50 | 0.00 | 0.50 | 0.00 | 0.50 | 0.00 | 0.50 | 0.00 |
| Max | 3.50 | 0.30 | 2.50 | 0.30 | 3.50 | 0.30 | 2.50 | 0.30 |
| Actual | 1.50 | 0.07 | 1.20 | 0.07 | 3.00 | 0.07 | 2.00 | 0.07 |
| After 1st stage | 1.42 | 0.12 | 1.12 | 0.01 | 2.68 | 0.06 | 2.13 | 0.07 |
| After 2nd stage | 1.50 | 0.07 | 1.20 | 0.07 | 3.00 | 0.07 | 2.00 | 0.07 |

granular strategy in the fuzzy case were assumed: a population size: 50, a number of generations: 75, a probability of the mutation: 0.2, a probability of the crossover 0.1.

The presented example was concerned with identification of uncertain geometrical granular parameters in the elastic system. In all three kinds of granularity the two stage strategy gave satisfactory results.

## 6.4 The Identification of Thermal Materials Parameters

A thermo-mechanical system presented in Figure 5 is considered [6]. One surface of the box is supported, whereas on the opposite one the point load is applied at every node (the total load is equal to 224kN). On the supported surface of the structure the temperature $T=10°C$ is applied. The third type thermal boundary condition (convection) is specified on the internal surface, where the ambient temperature $T^{\infty}$ and the heat convection coefficient $\alpha$ are identified.

The identification has been performed for 4 sensor points of temperature and 4 sensor points of displacements located on the external surfaces of the structure. The structure

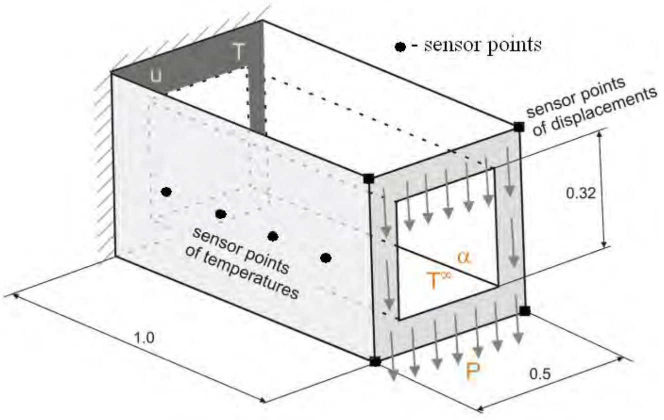


**Fig. 5.** Geometry, boundary conditions, location of the sensor points, distribution of the temperature and deformation of the structure

is made of steel whose material properties are identical as in the previous example. Figure 5 shows deformation and distribution of the temperature in the model.

The actual and found values of the heat convection coefficient $\alpha$ and ambient temperature $T^{\infty}$ are presented in the Table 15 (interval case), Table 16 (fuzzy case) and Table 17 (stochastic case).

The following parameters of the granular strategy in the interval and stochastic cases were assumed: a population size: 30, a number of generations: 100, a probability of the mutation: 0.2, a probability of the crossover 0,1. The following parameters of the granular strategy in the fuzzy case were assumed: a population size: 30, a number of generations: 200, a probability of the mutation: 0.2, probability of the crossover 0,1.

Identification of selected uncertain granular physical parameters of the thermo-mechanical system was considered. Similarly like in the previous examples the granular algorithm found actual parameters of the system.

**Table 15.** Results for interval identification

| | $\underline{\alpha}$ | $\overline{\alpha}$ | $\underline{T^{\infty}}$ | $\overline{T^{\infty}}$ |
|---|---|---|---|---|
| Min | 1.00 | 1.00 | 0.00 | 0.00 |
| Max | 25.00 | 25.00 | 105.00 | 105.00 |
| Actual | 3.00 | 7.00 | 45.00 | 52.00 |
| After 1st stage | 2.61 | 6.76 | 49.99 | 53.33 |
| After 2nd stage | 3.00 | 7.00 | 45.00 | 52.00 |

**Table 16.** Results for fuzzy identification (first and second alpha–cuts)

| alpha-cut | first | | | | second | | | |
|---|---|---|---|---|---|---|---|---|
| | $\underline{\alpha}$ | $\overline{\alpha}$ | $\underline{T^{\infty}}$ | $\overline{T^{\infty}}$ | $\underline{\alpha}$ | $\overline{\alpha}$ | $\underline{T^{\infty}}$ | $\overline{T^{\infty}}$ |
| Min | 1.00 | 1.00 | 0.00 | 0.00 | 1.00 | 1.00 | 0.00 | 0.00 |
| Max | 25.00 | 25.00 | 105.00 | 105.00 | 25.00 | 25.00 | 105.00 | 105.00 |
| Actual | 3.00 | 7.00 | 45.00 | 52.00 | 4.00 | 6.00 | 48.00 | 51.00 |
| After 1st stage | 2.05 | 8.45 | 43.54 | 54.43 | 4.97 | 6.45 | 49.94 | 51.93 |
| After 2nd stage | 3.00 | 7.00 | 45.00 | 52.00 | 4.00 | 6.00 | 48.00 | 51.00 |

**Table 17.** Results for stochastic identification

| | $\alpha$ | | $T^{\infty}$ | |
|---|---|---|---|---|
| | m | σ | m | σ |
| Min | 1.00 | 0.00 | 0.00 | 0.00 |
| Max | 25.00 | 5.00 | 100.00 | 20.00 |
| Actual | 5.00 | 0.70 | 50.00 | 10.00 |
| After 1st stage | 4.91 | 0.45 | 52.12 | 11.05 |
| After 2nd stage | 5.00 | 0.70 | 50.00 | 10.00 |

## 6.5 The Identification of Piezoelectric Parameters of Material

The identification of piezoelectric material constants and the polarization direction of a plate was considered [6]. The plate is subjected to a stress in *y*-direction and an applied voltage as shown in Figure 6. The PZT-4 ceramic material is modeled [6]. The applied voltage is $V_0$=1000 V. On the horizontal edges of the strip, the charge flux density is equal to zero. The applied stresses are equal to σ=5 MPa. The length of the strip is equal to L=1 mm, the height h=0.5 mm.

The 4 design variables are used: 3 piezoelectric constants and the value of the angle $\theta$, which describes the polarization direction. The displacements, electric potential and charge sensors are placed in every node.

The actual value and found value of piezoelectric parameters and polarization direction are presented in the Table 18 (interval case), Tables 19 and 20 (fuzzy case) and Table 21 (stochastic case).

The following parameters of the granular strategy in the interval and stochastic cases were assumed: a population size: 70, a number of generations: 140, a probability of the mutation: 0.2, a probability of the crossover 0,1. The following parameters of the granular strategy in the fuzzy case were assumed: a population size: 100, a number of generations: 200, a probability of the mutation: 0.2, a probability of the crossover 0,1.

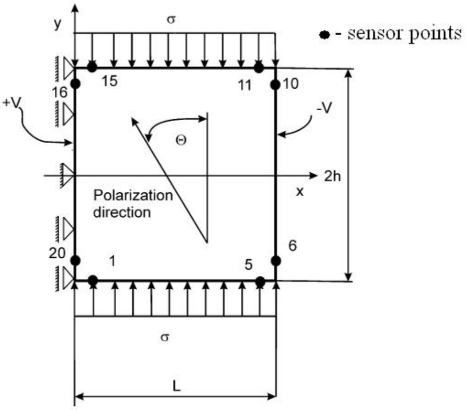


**Fig. 6.** Geometry and boundary conditions for the considered piezoelectric plate

**Table 18.** Results for interval identification

| | $\underline{e}_{11}$ | $\overline{e}_{11}$ | $\underline{e}_{13}$ | $\overline{e}_{13}$ | $\underline{e}_{33}$ | $\overline{e}_{33}$ | $\underline{\theta}$ | $\overline{\theta}$ |
|---|---|---|---|---|---|---|---|---|
| Min | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.00 | 0.00 |
| Max | 2.00 | 2.00 | 1.00 | 1.00 | 2.00 | 2.00 | 180.00 | 180.00 |
| Actual | 0.72 | 0.74 | 0.63 | 0.65 | 0.20 | 0.22 | 89.00 | 91.00 |
| After 1st stage | 0.70 | 0.73 | 0.61 | 0.65 | 0.26 | 0.27 | 88.54 | 90.34 |
| After 2nd stage | 0.72 | 0.74 | 0.63 | 0.65 | 0.20 | 0.22 | 89.00 | 91.00 |

**Table 19.** Results for fuzzy identification (first alpha-cut)

| | $\underline{e}_{11}$ | $\overline{e}_{11}$ | $\underline{e}_{13}$ | $\overline{e}_{13}$ | $\underline{e}_{33}$ | $\overline{e}_{33}$ | $\underline{\theta}$ | $\overline{\theta}$ |
|---|---|---|---|---|---|---|---|---|
| Min | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.00 | 0.00 |
| Max | 2.00 | 2.00 | 1.00 | 1.00 | 2.00 | 2.00 | 180.00 | 180.00 |
| Actual | 0.71 | 0.73 | 0.62 | 0.66 | 0.20 | 0.22 | 88.00 | 92.00 |
| After 1st stage | 0.63 | 0.76 | 0.60 | 0.68 | 0.16 | 0.25 | 88.78 | 92.56 |
| After 2nd stage | 0.71 | 0.73 | 0.62 | 0.66 | 0.20 | 0.22 | 88.00 | 92.00 |

**Table 20.** Results for fuzzy identification (second alpha-cut)

| | $\underline{e}_{11}$ | $\overline{e}_{11}$ | $\underline{e}_{13}$ | $\overline{e}_{13}$ | $\underline{e}_{33}$ | $\overline{e}_{33}$ | $\underline{\theta}$ | $\overline{\theta}$ |
|---|---|---|---|---|---|---|---|---|
| Min | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.00 | 0.00 |
| Max | 2.00 | 2.00 | 1.00 | 1.00 | 2.00 | 2.00 | 180.00 | 180.00 |
| Actual | 0.72 | 0.74 | 0.63 | 0.65 | 0.20 | 0.22 | 89.00 | 91.00 |
| After 1st stage | 0.66 | 0.69 | 0.61 | 0.65 | 0.18 | 0.21 | 89.73 | 91.98 |
| After 2nd stage | 0.72 | 0.74 | 0.63 | 0.65 | 0.20 | 0.22 | 89.00 | 91.00 |

**Table 21.** Results for stochastic identification

| | $e_{11}$ | | $e_{13}$ | | $e_{33}$ | | $\theta$ | |
|---|---|---|---|---|---|---|---|---|
| | m | σ | m | σ | m | σ | m | σ |
| Min | 0.20 | 0.00 | 0.20 | 0.00 | 0.10 | 0.00 | 0.00 | 0.00 |
| Max | 2.00 | 0.10 | 1.00 | 0.10 | 2.00 | 0.10 | 180.00 | 10.00 |
| Actual | 0.73 | 0.01 | 0.64 | 0.01 | 0.21 | 0.01 | 90.00 | 5.00 |
| After 1st stage | 0.70 | 0.00 | 0.59 | 0.03 | 0.25 | 0.04 | 92.32 | 3.01 |
| After 2nd stage | 0.73 | 0.01 | 0.64 | 0.01 | 0.21 | 0.01 | 90.00 | 5.00 |

The aim of identification was to find the granular physical parameters of the piezoelectric system. Optimal parameters of the granular optimization method provided the short time of computing.

### 6.6 The Identification of Laminate Materials Parameters

A rectangular simple laminate plate [3] made of the glass-epoxy is considered. Each ply of the symmetrical laminate has the same thickness $h_i$=0.002m. The stacking sequence of the laminate is: (0/45/90/-45/0/90/0/90)s [3]. The sensor points are located on the edges of the plate.

The actual and found values of laminate parameters are presented in the Table 22 (interval case), Tables 23 and 24 (fuzzy case) and Table 25 (stochastic case).

The following parameters of the granular strategy in the interval and stochastic cases were assumed: a population size: 100, a number of generations: 200, a probability of the

**Table 22.** Results for interval identification

| | $\underline{E}_1$ | $\overline{E}_1$ | $\underline{E}_2$ | $\overline{E}_2$ | $\underline{\nu}_{12}$ | $\overline{\nu}_{12}$ | $\underline{G}_{12}$ | $\overline{G}_{12}$ |
|---|---|---|---|---|---|---|---|---|
| Min | 1.92E10 | 1.92E10 | 4.20E09 | 4.20E09 | 0.190 | 0.190 | 9.70E08 | 9.70E08 |
| Max | 5.08E10 | 5.08E10 | 1.08E10 | 1.08E10 | 0.410 | 0.410 | 8.03E09 | 8.03E09 |
| Actual | 3.70E10 | 4.00E10 | 8.00E09 | 8.50E09 | 0.250 | 0.270 | 4.00E09 | 4.20E09 |
| After 1st stage | 3.72E10 | 3.99E10 | 8.07E09 | 8.67E09 | 0.243 | 0.272 | 4.01E09 | 4.24E09 |
| After 2nd stage | 3.70E10 | 4.00E10 | 8.00E09 | 8.50E09 | 0.250 | 0.270 | 4.00E09 | 4.20E09 |

**Table 23.** Results for fuzzy identification (first alpha-cut)

| | $\underline{E}_1$ | $\overline{E}_1$ | $\underline{E}_2$ | $\overline{E}_2$ | $\underline{\nu}_{12}$ | $\overline{\nu}_{12}$ | $\underline{G}_{12}$ | $\overline{G}_{12}$ |
|---|---|---|---|---|---|---|---|---|
| Min | 1.92E10 | 1.92E10 | 4.20E09 | 4.20E09 | 0.190 | 0.190 | 9.70E08 | 9.70E08 |
| Max | 5.08E10 | 5.08E10 | 1.08E10 | 1.08E10 | 0.410 | 0.410 | 8.03E09 | 8.03E09 |
| Actual | 3.82E10 | 3.90E10 | 8.23E09 | 8.31E09 | 0.257 | 0.263 | 4.10E09 | 4.18E09 |
| After 1st stage | 3.89E10 | 3.92E10 | 8.20E09 | 8.31E09 | 0.258 | 0.264 | 4.09E09 | 4.19E09 |
| After 2nd stage | 3.82E10 | 3.90E10 | 8.23E09 | 8.31E09 | 0.257 | 0.263 | 4.10E09 | 4.18E09 |

**Table 24.** Results for fuzzy identification (second alpha-cut)

| | $\underline{E}_1$ | $\overline{E}_1$ | $\underline{E}_2$ | $\overline{E}_2$ | $\underline{\nu}_{12}$ | $\overline{\nu}_{12}$ | $\underline{G}_{12}$ | $\overline{G}_{12}$ |
|---|---|---|---|---|---|---|---|---|
| Min | 1.92E10 | 1.92E10 | 4.20E09 | 4.20E09 | 0.190 | 0.190 | 9.70E08 | 9.70E08 |
| Max | 5.08E10 | 5.08E10 | 1.08E10 | 1.08E10 | 0.410 | 0.410 | 8.03E09 | 8.03E09 |
| Actual | 3.84E10 | 3.87E10 | 8.25E09 | 8.28E09 | 0.259 | 0.261 | 4.12E09 | 4.15E09 |
| After 1st stage | 3.87E10 | 3.92E10 | 8.26E09 | 8.29E09 | 0.258 | 0.262 | 4.12E09 | 4.18E09 |
| After 2nd stage | 3.84E10 | 3.87E10 | 8.25E09 | 8.28E09 | 0.259 | 0.261 | 4.12E09 | 4.15E09 |

**Table 25.** Results for stochastic identification

| | $E_1$ | | $E_2$ | | $\nu_{12}$ | | $G_{12}$ | |
|---|---|---|---|---|---|---|---|---|
| | m | σ | m | σ | m | σ | m | σ |
| Min | 2.00E10 | 0.00E9 | 4.00E9 | 0.00E9 | 0.000 | 0.000 | 2.00E9 | 0.10E8 |
| Max | 6.00E10 | 0.30E9 | 9.00E9 | 0.30E9 | 0.500 | 0.100 | 6.00E9 | 0.70E8 |
| Actual | 3.86E10 | 0.12E9 | 8.28E9 | 0.20E9 | 0.260 | 0.020 | 4.14E9 | 0.50E8 |
| After 1st stage | 3.92E10 | 0.11E9 | 8.14E9 | 0.17E9 | 0.270 | 0.040 | 4.07E9 | 0.22E8 |
| After 2nd stage | 3.86E10 | 0.12E9 | 8.28E9 | 0.20E9 | 0.260 | 0.020 | 4.14E9 | 0.50E8 |

mutation: 0.2, a probability of the crossover 0,1. The following parameters of the granular strategy in the fuzzy case were assumed: a population size: 200, a number of generations: 200, a probability of the mutation: 0.2, a probability of the crossover 0,1.

The presented example was concerned with identification of uncertain granular material parameters of laminates. In presented kinds of granularity (interval, fuzzy and stochastic) the two stage strategy gave satisfactory results.

## 7 Conclusions

An effective intelligent technique based on the granular evolutionary computing has been presented. This approach can be applied in optimization and identification of physical systems that are in the uncertain conditions. Various physical systems as: elastic, elastoplastic, thermo-mechanical, piezoelectric and composites were considered. Moreover the presented methodology can be extended also to biophysics systems.

Three models of granularity were examined: (i) interval, (ii) fuzzy and (iii) stochastic. Applications of other kinds of granularity based on e.g. rough sets, can be also possible [13][24].

The granular evolutionary algorithm aided by the local gradient method supported by the artificial neural network has turned out as an efficient approach. In all examined examples of identification the granular computing approach enables to find unknown parameters of systems with the great precision. This approach can be also very promising for reliability optimization in which the safety of a system is estimated and represented by a certain measure of uncertainty of its failure.

The crucial problem is selection of a suitable model of granularity. The stochastic approach is very useful and convenient in the case of existing of statistical data. When experimental data is obtained only from a few measurements and the probability density function is unknown the interval approach is much more convenient. If parameters of the systems are evaluated on the basis of some linguistic descriptions [24] then the fuzzy approach of granularity is more preferred.

The guiding principle of granular computing in identification problems is to exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve robustness and low solution cost.

**Acknowledgements.** The work was done as a part of project N502 4573 33 sponsored by the Polish Ministry of Science and Higher Education.

## References

1. Arbib, M.A. (ed.): The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge (1995)
2. Bargiela, A., Pedrycz, W.: Granular Computing: An introduction. Kluwer Academic Publishers, Boston (2002)
3. Beluch, W., Burczyński, T., Orantek, P.: The Two-Stage Fuzzy Strategy in identification of laminates elastic constants. In: The Proceedings of the CMM 2007 Conference, Lodz-Spala (2007)
4. Bui, H.D.: Inverse Problems in the Mechanics of Materials: An Instroduction. CRC Press, Bocca Raton (1994)

5. Burczyński, T., Beluch, W., Długosz, A., Orantek, P., Skrobol, A.: Inteligent computing in inverse problems. Computer Assisted Mechanics and Engineering Sciences 13, 161–206 (2006)
6. Burczyński, T., Długosz, A., Dziatkiewicz, G., Orantek, P.: Identification of the uncertain parameters in coupled problems. In: The Proceedings of the CMM 2007 Conference, CD-Edition, Lodz-Spala (2007)
7. Burczynski, T., Orantek, P., Kuś, W., Górski, R., Poteralski, A., Szczepanik, M.: The identification of uncertain parameters in mechanical structures. In: The Proceedings of the CMM 2007 Conference, CD-Edition, Lodz-Spala (2007)
8. Burczyński, T., Skrzypczyk, J.: Fuzzy aspects of the boundary element method. Engineering Analysis with Boundary Elements 19, 209–216 (1997)
9. Burczyński, T., Skrzypczyk, J.: Theoretical and Computational Aspects of the Stochastic Boundary Element Method. Computer Methods in Applied Mechanics and Engineering 168, 321–344 (1999)
10. Cordon, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems:current framework and new trends. Fuzzy Sets and Systems 141, 5–31 (2004)
11. Cordon, O., Herrera, F., Hoffmann, F., Magdalena, L.: Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases. World Scientific, Singapore (2001)
12. Czogała, E., Pedrycz, W.: Elements and Methods of Theory of Fuzzy Sets. PWN, Warsaw (1985)
13. Dubois, D., Prade, H.: Fuzzy Sets and Systems. Academic Press, New York (1988)
14. Kleiber, M., Hien, T.D.: Stochastic Finite Element Method. John Willey & Sons, New York (1992)
15. Littman, M., Ackley, D.: Adaptation in constant utility nonstationary environment. In: Proc. International Conference on Genetic Algorithms (1994)
16. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolutionary Programs. Springer, Berlin (1996)
17. Moore, R.E., Kearfott, R.B., Cloud, M.J.: Introduction to Interval Analysis. SIAM Press, Philadelphia (2009)
18. Orantek, P.: An intelligent computing technique in identification problems. Computer Assisted Mechanics and Engineering Sciences 13, 351–364 (2006)
19. Papoulis, A.: Probability, Random Variables, and Stochastic Processes. McGraw Hill, New York (1991)
20. Pedrycz, W.: Fuzzy evolutionary computing. Soft Computing 2 (1998)
21. Pedrycz, W.: Computational Intelligence: An Introduction. CRC Press, Boca Raton (1997)
22. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordrecht (1991)
23. Polkowski, L., Skowron, A. (eds.): Rough Sets in Knowledge Discovery. Methodology and Applications, vol. I. Physica-Verlag, Heidelberg (1998)
24. Zadeh, L.A., Kacprzyk, J.: Computing with Words in Information/IntelligentSystems, vol. 1-2. Physica-Verlag, Heidelberg (1999)
25. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)

# Visualizing Huge Image Databases by Formal Concept Analysis

Kazuhito Sawase[1], Hajime Nobuhara[1], and Barnabas Bede[2]

[1] Department of Intelligent Interaction Technologies,
University of Tsukuba, Tenoudai 1-1-1, Tsukuba science city,
Ibaraki 305-8573, Japan
{sawase,nobuhara}@cmu.iit.tsukuba.ac.jp

[2] Department of Mathematics, University of Texas- Pan American
Edinburg, Tx, 78539-2999, USA
bede.barna@bgk.bmf.hu

**Abstract.** Based on formal concept analysis we propose a novel lattice visualization system for huge image databases as a realization of the important paradigm of human-centered information processing based on granular computing. From a given cross table of objects (images) and attributes (image features) the proposed system first constructs a concept lattice. Then the Hasse diagram of this lattice is visualized. The information granules in the proposed system correspond to the elements of the concept lattice. All the important components of granular computing are shown to be present in the proposed system, such as: abstraction of data, derivation of knowledge and empirical verification of the abstraction. Since formal concept analysis generates an order relation, we obtain a hierarchical structure of concepts. This structure is shown to be also strongly related to the granular computing, since this is how the lattice visualization system implements the zoom in and zoom out capability of granular computing systems. Using the proposed system, a user can freely analyze the perspective and detailed structure of a large image database in the setting of granular computing. Furthermore, through an interaction function, the potential user can adjust the quantization of features, being able in this way, to select the attributes which allow him to obtain a suitable concept lattice. Therefore, the proposed system can be regarded as a promising human-centric information processing algorithm, based on granular computing.

## 1 Introduction

Due to the spread of high performance computers, digital input devices and high capacity memory devices, information explosion is becoming nowadays a more and more acute problem raised by the society to the information processing community [26]. Especially the amount of digital images has been rapidly increasing in accordance with prevailing digital cameras for individual use.

IDC [25] reports that the digital universe was in 2007, at the level of 281 exabytes or 281 billion gigabytes. Meanwhile, the ability of human-beings to process the information has not dramatically improved in the last several hundreds of years. Due to this information explosion, we need novel methods to

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 351–373.
springerlink.com

visualize information contained in image databases or video sequences (see e.g., [5] [6] [8] [20] [27]).

In order to extract information with understandable contents and suitable size from such a huge volume of data, we need new, intelligent, human-centric information processing techniques and information visualization schemes. The users need to be able to recognize and understand comprehensive information about several images at once.

### 1.1 Information Visualization

In the topic of information visualization there is a very intense, ongoing research. Almost all the research efforts in the topic of information visualization [5] [6] [8] [20] [27], with target objects either text or images are directed towards a metric spaces based approach, where the similarity between objects is obtained based on a distance between them. From the mathematical point of view, this research can be sought as metric structure based visualization.

This chapter introduces a novel idea in this research field, by proposing a visualization method for huge image databases based on *granular computing* [1] [23], and *formal concept analysis* [9].

### 1.2 Granular Computing

Human centric information processing traces back to the celebrated works of L. Zadeh [22], [23]. Indeed, Information Granulation appears for the first time in the paper [23], but as Zadeh himself underlines in [23], the basic ideas related to this concept can be rooted back to the very beginning of Fuzzy Sets Theory, [22]. Human-centric information processing relates to the concept of information granulation since this is one of the three basic tasks underlying human cognition [23]. Information Granulation is the conceptual framework for granular computing. The Granular Computing paradigm was first considered by T.Y. Lin in [13] and the first monograph in this direction was written by A. Bargiela and W. Pedrycz, [1].

Information granulation (see e.g., [21], [23], [24] ) is the process of grouping elements based on their indistinguishability, similarity, proximity or functionality. According to [1], information granules are complex entities that arise in the process of abstraction of data and derivation of knowledge from information. Also, as it is shown in [1], information granules typically have a hierarchical structure. From a mathematical point of view, this is shown in the very recent paper [2], to be the Information Sciences analogue to the axiomatization of the Classical Set Theory using classes (Von Neumann–Bernays–Gödel set theory). This idea and the mathematical formulation of the Von Neumann–Bernays–Gödel set theory leads to a higher level of abstraction and generalization in this area, which further emphasizes the hierarchical nature of granular computing. Following this definition for information granules, granular computing can be defined as (see [2]) a “structured combination of algorithmic abstraction of data and nonalgorithmic empirical verification of the semantics of abstractions”. This definition allows a higher level of generality since there are no prescribed mechanisms to perform the tasks of abstraction of data and derivation of knowledge (see also [14]) and it is consistent with the hierarchical approach of [1].

### 1.3 Granular Computing in Image Processing

Human beings have the gift that looking at an image or a set of images, they are able immediately to perform the granulation of data and derivation of knowledge. For machines this is not at all a trivial task and the way how information granulation is performed is usually application oriented, despite the trend towards generality in what regards this issue. In the Image Processing field the idea of human centered information processing and granularity of information was present tacitly for a long time ([1]), since the pattern recognition field has its central issue the recognition, classification and abstraction of image data. It appeared recently also explicitly in e.g. ([1], [15], [16], [19]).

Information in Image Processing is represented by images. Machines represent them as matrices with integer elements. Human beings, as mentioned earlier have the ability to see in an image beyond the numbers a “flower” or a “dinosaur”. Our point is that an information granule that humans use for the representation of a “flower” is a collection of images gathered by some image features and attributes. This is also consistent with the ideas and the formalism in [2]. Indeed we can regard images as information (set) and collections of several images, together with their features as information granules (classes).

### 1.4 Ordered Structure-Based Image Processing

Furthermore, our point is that in the visualization problem, using only the idea of “distance” is not precisely reflecting the human way of thinking. Indeed, human beings can recognize a flower in an image that contains also buildings and animals. This gift of human beings translates in mathematical terms into an order relation, “inclusion” relation between objects, in some sense. In terms of mathematical structures this pushes our thoughts from a purely metric spaces based approach, towards an ordered structure based approach for image processing.

Most of the literature in information visualization focuses on a metric space approach so conventional image database visualization systems are unable to show the perspective nature of the whole database. The present chapter, in contrast, focuses on a different kind of basic mathematical structure, i.e., the ordered structure [7] [11]. This approach allows us to develop and to implement a more comprehensive information visualization strategy, namely, we visualize the lattice structure of an image database. It is easy to see that the ordered structure is very suitable for human perception and intuition. Partially this is the reason why fuzzy sets theory, based on a lattice structure as well, is able to interpret and to implement expert’s knowledge [12] [17] [22]. Consequently, the ordered structure has a great potential ability to fit our intent to perform information processing tasks in a more and more human-centered manner. This approach allows us to deal with objects and relations in terms of an *ordered* *structure* [7] [11], which is different from the mathematical structure used by the conventional ongoing research.

### 1.5 Visualizing Huge Image Databases by Formal Concept Analysis

Formal concept analysis [9] constructs a concept lattice (which is a complete lattice) from the context table (information table) which contains objects and their attributes (feature vectors). The concept lattice has an ordered structure induced by the order of different concepts in a given context. This structure is very intuitive and so it is suitable for human perception. The proposed system is able to visualize a complete lattice obtained from a huge image database or a long video sequence, and users can easily recognize, understand and moreover, they can further process comprehensive information about the images in the database at once. In the present paper we discuss in detail the construction process which generates the context table, the quantization of attributes, and finally the generation of the concept lattice itself.

Finally the Hasse diagram of the concept lattice is visualized, and it is easy to see that it supports the empirical verification of the semantics of abstractions. Also, the knowledge obtained automatically is intuitively communicated to the user. So, a huge image database is visualized by the proposed method at once. Surely the hierarchical structure of the granular information makes it possible to the user to zoom in and out at any information granule.

The proposed system was realized using the JAVA based programming language *Processing* [18], on an ordinal platform (CPU = 2.13GHz, MM = 2GB). As visualization experiments, we perform lattice visualization for the Corel Image Gallery and the Ubiquitous Home Image Database.

Since concept lattices are strongly related to the processes of discretization and quantization, we can see that the proposed visualization system perfectly fits in the setting of granular computing [1]. Indeed, we can see in the proposed system all the important elements of granular computing. The link between the proposed system and granular computing based human-centric information processing can be summarized as follows:

- Abstraction of data is performed through the discretization and quantization of features and attributes. Then a relation which relates objects to attributes is constructed. This way we obtain a huge number of abstract data as objects and very complex relations.
- Derivation of knowledge, is performed through construction of a complete lattice by formal concept analysis.
- Formal concept analysis is based on an ordered structure given by the set inclusion between attributes as order relation. This hierarchical structure of concepts in the proposed structure corresponds exactly to the hierarchical representation of granular elements. Using this ordered structure, we have realized the zoom in and zoom out capability of granular computing.
- The binary relation used in the formal concept analysis can be extended to a fuzzy relation, i. e., the proposed system can naturally work in the context of human subjectivity. [3]

- The ideas underlying the visualization platform and the interaction function of the proposed system give a great ability to support the empirical verification of the results. Moreover, the potential user can adjust the quantization of features, being able in this way, to select the attributes which allow the him to obtain a suitable concept lattice. This interaction between user and machine through the formal concept analysis can be regarded as a promising human-centric, granular computing-based information processing method.

In Sec. 2 give some preliminaries about formal concept analysis. In Sec. 3, we present the proposed formal concept analysis-based lattice visualization system, aimed to allow users to understand and extract information from large image databases or video sequences. Section 4 shows two visualization experiments of two huge image databases: the Corel Image Gallery and Ubiquitous Home Database produced by National Institute of Information and Communication Technology (NICT). In Sec. 5, we explain the relationship between the proposed lattice visualization system and the important paradigms of granular computing and in Sec. 6 we conclude the paper.

## 2 Formal Concept Analysis

### 2.1 Formal Concept Analysis – An Overview

Formal concept analysis [7] [9] is a powerful mathematical tool, which helps us to construct a complete lattice from a cross table (relation) (e.g. Tab. 1) between of objects and attributes. Let us illustrate first on an example, how we can use formal concept analysis in our problem. We construct in the followings a complete lattice from objects and attributes corresponding respectively to images and features.

In Tab. 1, the fact that the $i$th object possesses the $j$th attribute is indicated by the symbol × in the $ij$-position of the table, and Figs. 1 - 4 correspond to the images 1 - 8.

In this context, a concept will be an ordered pair $(A, B)$, where $A$ (the extent) is a subset of a set consisting of eight images and $B$ (the intent) is a subset of the

**Table 1.** An example of an information table

| | Human | Animals | RED | GREEN | BLUE | WHITE |
|---|---|---|---|---|---|---|
| Image 1 | × | | | | × | × |
| Image 2 | | | | | × | × |
| Image 3 | | × | | × | | |
| Image 4 | | × | | | | × |
| Image 5 | | | × | × | | |
| Image 6 | | | × | × | | |
| Image 7 | | × | | × | | |
| Image 8 | | | | | × | × |

**Fig. 1.** Example: Images 1 and 2

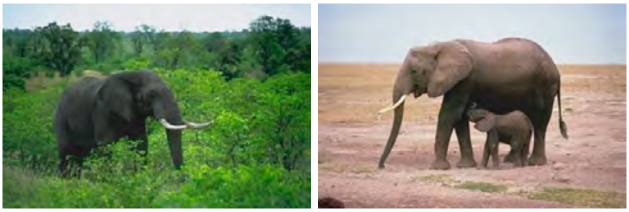


**Fig. 2.** Example: Images 3 and 4

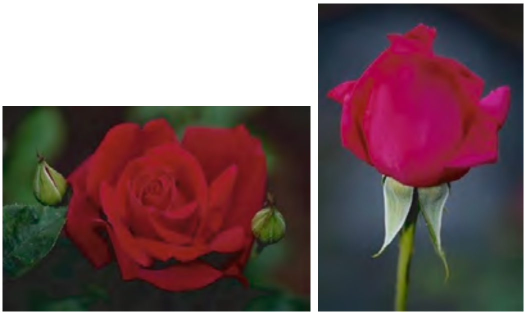


**Fig. 3.** Example: Images 5 and 6

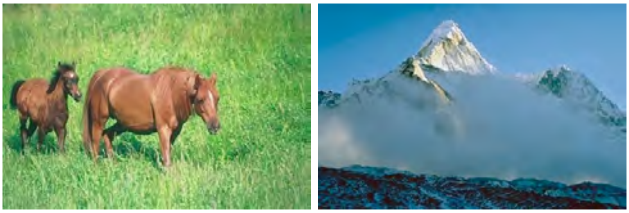


**Fig. 4.** Example: Images 7 and 8

six types of image features, e.g., "Human" means that there are human beings in the image. To demand that the concept is determined by its extent and by its intent, means that $B$ should contain just the properties shared by all the images in $A$ and, similarly, the images in $A$ should be precisely those sharing all the properties in $B$. A simple procedure for finding a concept is as follows: take an object, say the "Image 2", and let $B$ be the set of attributes which it possesses, in this case

$$B = \{\text{BLUE, WHITE}\}. \tag{1}$$

Let $A$ be the set of all frames possessing all the attributes in $B$,i.e., in our case

$$A = \{\text{Image 2, Image 8}\}. \tag{2}$$

Then $(A, B)$ is a concept, and in this context, we can interpret this concept

$$(A, B) = (\{\text{Image 2, Image 8}\}, \{\text{BLUE, WHITE}\}), \tag{3}$$

as a "resort", since the images "Image 2" and "Image 8" are representing 'beach' and 'mountain', respectively. In other words, the concept 'resort' always contains the attributes 'BLUE' and 'WHITE'. If we will have any images with attributes 'BLUE' and 'WHITE', we can guess that the image is related to 'resorts' based on this relationship (knowledge). More generally, we may begin with a set of objects rather than a single object. Concepts may also be obtained via a similar process commencing with a set of attributes.

It is usual to regard a concept $(A_1, B_1)$ as being 'less general' than a concept $(A_2, B_2)$ if the extent $A_1$ of $(A_1, B_1)$ is contained in the extent $A_2$ of $(A_2, B_2)$. Thus an order is defined on the set of concepts by

$$(A_1, B_1) \leq (A_2, B_2) \quad \Leftrightarrow \quad A_1 \subseteq A_2. \tag{4}$$

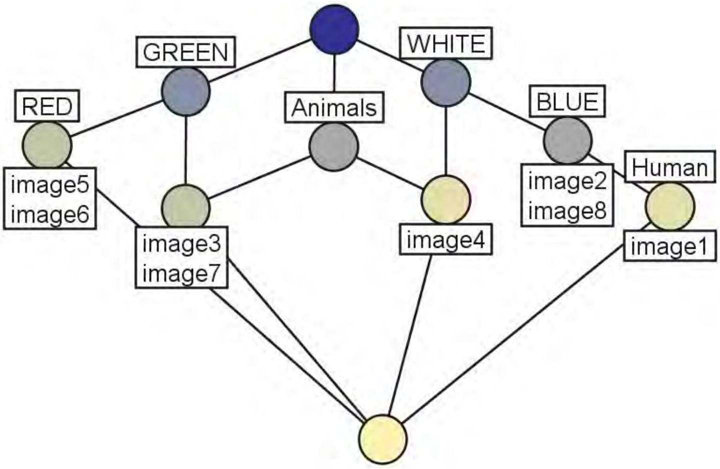


**Fig. 5.** Concept Lattice with respect to Table 1

The asymmetry in this definition is illusory since $A_1 \subseteq A_2$ is equivalent to $B_1 \supseteq B_2$. The resulting ordered set of concepts for the image database is the lattice given in Fig. 5.

### 2.2 An Intuitive Algorithm to Construct the Concept Lattice

We show in the present section how to construct a concept lattice from any cross table, based on the following precise definitions. A context is a triple $(G, M, I)$ where $G$ and $M$ are two sets and $I \subseteq G \times M$ is a relation. The elements of $G$ and $M$ are called **objects** and **attributes** respectively. As it is usual, instead of writing $(g, m) \in I$, we write $gIm$ and we say that 'the object $g$ has the attribute $m$'. For $A \subseteq G$ and $B \subseteq M$, we define

$$A' = \{m \in M \mid (\forall g \in A)\ gIm\}, \tag{5}$$

$$B' = \{g \in G \mid (\forall m \in B)\ gIm\}, \tag{6}$$

so $A'$ is the set of attributes common to all the objects in $A$ while $B'$ is the set of objects possessing the attributes in $B$. Then a **concept** of the context $(G, M, I)$ is defined to be a pair $(A, B)$ where $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. The **extent** of the concept $(A, B)$ is $A$ while its intent is $B$. The set of all concepts of the context $(G, M, I)$ is denoted by $\mathbf{B}(G, M, I)$.

Let $(G, M, I)$ be a context. For concepts $(A_1, B_1)$ and $(A_2, B_2)$ in $\mathbf{B}(G, M, I)$ we write $(A_1, B_1) \leq (A_2, B_2)$, if $A_1 \subseteq A_2$. Also, $A_1 \subseteq A_2$ implies that $A_1^{'} \supseteq A_2^{'}$, and the reverse implication is valid too, because $A_1^{''} = A_1$ and $A_2^{''} = A_2$. We have therefore,

$$(A_1, B_1) \leq (A_2, B_2) \ \Leftrightarrow\ A_1 \subseteq A_2 \ \Leftrightarrow\ B_1 \supseteq B_2. \tag{7}$$

We can easily see that the relation $\leq$ is an order relation on $\mathbf{B}(G, M, I)$, and $\langle \mathbf{B}(G, M, I); \leq \rangle$ is a complete lattice, i.e., it is the **concept lattice** of the context $(G, M, I)$.

We consider in what follows an intuitive algorithm for the construction of the concept lattice. First, we select an object of the context $(G, M, I)$, and then using Eqs. (5) and (6), we can successively find the corresponding concepts.

The detailed procedure is as follows:

1. **Find all the extents of the concepts in the context** $(G, M, I)$
   a) Draw a table with two columns headed **attributes** and **extents**. Leave the first cell of the attributes column empty and write $G$ in the first cell of the extents column.
   b) Find a maximal attribute-extent, say $m'$.
      i. If the set $m'$ is not already in the extents' column, add the row $[m|m']$ to the attribute-extent table. Intersect the set $m'$ with all previous extents in the Extents column. Add these intersections to the Extents' column (unless they are already in the list), and leave the corresponding cells in the Attribute column empty.

   ii. If the set $m'$ is already in the Extents column, add the label $m$ to the attribute cell of the row where $m'$ previously occurred.
  c) Delete the column below $m$ from the table.
  d) If the last column has been deleted, stop, otherwise return to 1-(b).

2. **Draw the diagram with $m$ and $m'$ labels.** Start at the top of the diagram with one point labeled $G$. Work down the list of Extents in the table from 1. For each set $S$ in the list, add an appropriately positioned new point to the diagram. Below the point corresponding to $S$ list the elements in $S$. If $S$ is an attribute-extent, say $S = m'$, add the label $m$ above the point corresponding to $S$.
3. **Redraw the diagram with $g$ and $m$ labels**
  a) Redraw the diagram. Add the $m$ labels as in the first diagram.
  b) For each object $g$ in $G$, add a label $g$ below the point on the diagram which has the smallest extent containing the object $g$ (this point can be found from the first diagram). Alternatively, the point $g$ to be labeled can be obtained by finding the point $\bigwedge\{m \mid gIm\}$.
4. **Check the answer**.
  a) Check that every joint-irreducible element has a label $g \in G$.
  b) Check that every meet-irreducible element has a label $m \in M$.
  c) Check that

$$(\forall g \in G)(\forall m \in M)gIm \ \Leftrightarrow g \leq m \tag{8}$$

  by checking that, for all $m \in M$, the set of object labels in $\downarrow m$ is exactly the attribute-extent $m'$, where $\downarrow m$ stand for the down set of $m$ i.e., the set of all elements less than $m$.

## [Example]

Using the cross table shown in Table 1, we present in this particular case the way how we can construct the concept lattice. In order to improve the visualization of the lattice, we convert Table 1 into the following table 2. With respect to Table 2, we apply the intuitive algorithm shown above, and we obtain the attributes-extents table given in Tab. 3.

**Table 2.** A cross table

| | Hu | An | Re | Gr | Bl | Wh |
|---|---|---|---|---|---|---|
| 1 | × | | | | × | × |
| 2 | | | | | × | × |
| 3 | | × | | × | | |
| 4 | | × | | | | × |
| 5 | | | × | × | | |
| 6 | | | × | × | | |
| 7 | | × | | × | | |
| 8 | | | | | × | × |

**Table 3.** Attributes and Extents Table

| Attributes | Extents |
|---|---|
| | G |
| Gr | $\{3, 5, 6, 7\}$ |
| Wh | $\{1, 2, 4, 8\}$ |
| An | $\{3, 4, 7\}$<br>$\{3, 7\}$<br>$\{4\}$ |
| Bl | $\{1, 2, 8\}$ |
| Re | $\{5, 6\}$ |
| Hu | $\{1\}$ |
| | $\phi$ |

We observe the inclusions in Tab. 3, and so we obtain Fig. 5, where the overlaps are eliminated.

### 2.3 Next Closure Algorithm

The intuitive algorithm shown in the previous subsection includes some redundant computations, therefore in the present chapter we will use the **next closure algorithm** to perform the formal concept analysis step [9]. We will briefly describe the next closure algorithm in the followings:

**[Def. Lexicographic Order]**

$A, B \subseteq G$, $i \in G$
i)

$$\begin{aligned} A < B \quad &\Leftrightarrow \quad \exists i \in B \setminus A, \\ A \cap \{1, 2, \ldots, i-1\} \ &= \ B \cap \{1, 2, \ldots, i-1\}. \end{aligned} \tag{9}$$

ii)

$$\begin{aligned} A <_i B \quad &\Leftrightarrow \quad \in B \setminus A, \\ A \cap \{1, 2, \ldots, i-1\} \ &= \ B \cap \{1, 2, \ldots, i-1\}. \end{aligned} \tag{10}$$

**[Def. Next Closure]**

$$A \oplus i = ((A \cap \{1, \ldots, i-1\}) \cap \{i\}) \, . \tag{11}$$

According to the above definitions, the next closure algorithm will work as follows:

**[Next Closure Algorithm]**

```
main
  begin
    $A = \phi''$
  repeat
    Output A
    Next Closure
  until not success
  end
```

## 3 A Lattice Visualization System for Huge Image Databases

As shown in the previous section, we can construct concept lattices from any cross tables having texts, images, video or sound files as objects, provided that they have precisely defined attributes. The main, still not completely solved question is, how to quantize the feature vectors to obtain relevant but not redundant attributes. These can contain multiple values as e.g., intensities of pixels in $\{0, 1, \ldots, 255\}$) while, in contrast, the attributes should be binary values. Of course, we can assign the digit of each feature value to an attribute in the cross table, however, the concept lattice will be too complex if we adopt such a strategy. The quantization should be performed carefully, however, we currently do not have any systematic way to quantize the feature vectors into attributes. This is apparently a drawback but it can be turned into an advantage allowing a more human-centered processing. Therefore, we have developed a lattice structure visualization system with an interaction function, where the user can adjust the

```
Next Closure
  begin
    i = maximum element of G
    i= succ(i)
    success = false
  repeat
    i = pred(i)
    if $(i \notin A)\{$
    $B = (A \cap \{1, 2, \ldots, i-1\}) \cap \{i\}$
    $C = B''$
    if$(A <_i C)\{$
    A = C
    success = true
    }
    }
  until success or i = minimum element of G
```

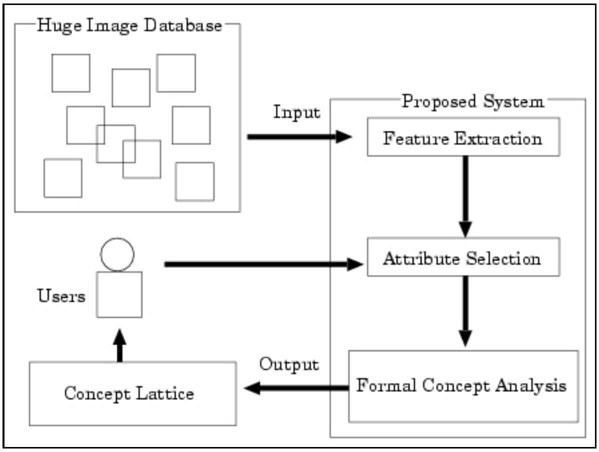


**Fig. 6.** Overview of the proposed visualization system

way how the features of images (color, intensity information) are quantized into attributes (Fig. 6).

The proposed system defines the features of images according to the following approach. We denote the images by $P_k$ $(k = 1, 2, \ldots, P_{max})$ ($P_{max}$ = the number of images in database) as the objects $G = \{P_k | k = 1, 2, \ldots, P_{max}\}$ of the cross table. Each image of the size $m \times n$ is represented by three image planes, i.e., read, green, blue, therefore, the definition of an object is as follows:

$$P = \left\{ P_k^c \mid P_k^c \in \{0, \ldots, 255\}^{m \times n}, c \in \{R, G, B\} \right\}. \tag{12}$$

In order to acquire a suitable and at the same time understandable lattice structure for the proposed human visualization system, HSB color space is employed. Furthermore, as attributes used in formal concept analysis, we propose new image features considering the effects of saturation and brightness.

The Hue (H), Saturation (S), and Brightness (B) are respectively defined as

$$H_k(i,j) = \tan^{-1} \frac{\gamma}{\beta - \alpha}, \tag{13}$$

$$S_k(i,j) = \frac{\alpha^2 + \beta^2 + \gamma^2}{3} \tag{14}$$

and

$$B_k(i,j) = P_k^R(i,j) + P_k^G(i,j) + P_k^B(i,j), \tag{15}$$

where

$$\begin{cases} \alpha = P_k^B(i,j) - P_k^R(i,j), \\ \beta = P_k^R(i,j) - P_k^G(i,j), \\ \gamma = P_k^G(i,j) - P_k^B(i,j). \end{cases} \tag{16}$$

In the framework of the new image features that we are currently using in formal concept analysis, the weights of Saturation, and Brightness are given by

$$S_w(P_k(i,j)) = \frac{S(P_k(i,j))}{S_{max}}, \tag{17}$$

and

$$B_w(P_k(i,j)) = 1 - \frac{|B(P_k(i,j)) - \frac{B_{max}}{2}|}{\frac{B_{max}}{2}}, \tag{18}$$

where $S_{max}$ and $B_{max}$ are the maximum values of the Saturation and Brightness respectively.

Finally we define a modified Hue as a new image feature as

$$H'(P_k(i,j)) = \sum_{0 \leq i \leq m, 0 \leq j \leq n} H(P_k(i,j)) S_w(P_k(i,j)) B_w(P_k(i,j)). \tag{19}$$

The attribute sets used in the framework of formal concept analysis are defined as

$$M = \{A_1, A_2, \ldots, A_{Max}\}, \tag{20}$$

where

$$A_k = \begin{cases} 0 & otherwise, \\ 1 & T_{k-1} \leq H' < T_k, \end{cases} \tag{21}$$

and $T$ is a threshold value. The index 'max' corresponds to the quantization number of our color space, and the proposed system can adjust this value through computer-user interaction.

## 4 Visualization Experiments by the Proposed System

The proposed visualization system is developed on a usual platform (CPU = 2.13GHz, MM = 2GB) using the JAVA based programming language 'Processing' [18].

In the first step of the algorithm, the proposed system successively loads all the images from a given database. Then, with respect to each image we perform the image feature extraction and so we generate the candidates for the attributes used in the formal concept analysis step. In the next step, the user sets the desired attributes by the interaction function proposed, and the concept lattice will be generated based on the selected attributes.

### 4.1 Visualization of Corel Image Database

For our first experiment we use the Corel Image Gallery which consists of 1,000 images, shown in Fig. 7, in the categories of *Human, Dinosaur, Flower, Elephant, Bus, Mountain, Horse, Dish, Building* and *Sea.* This visualization experiment

**Fig. 7.** Examples of Corel Image Database

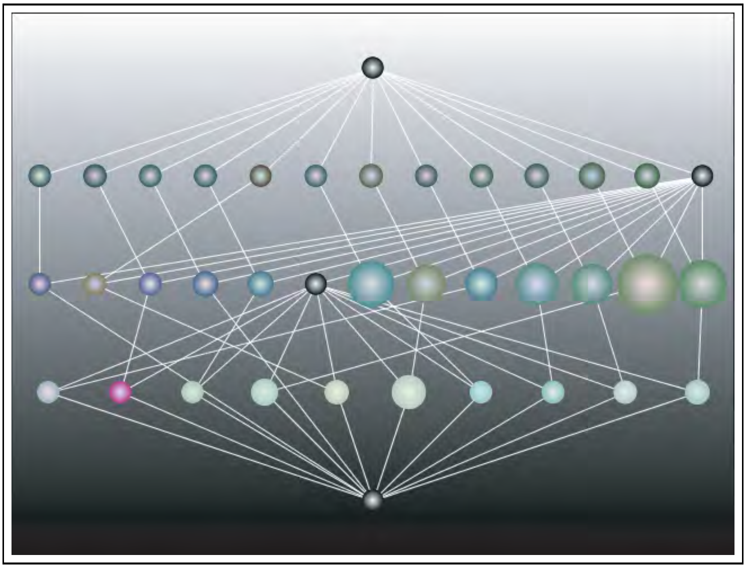


**Fig. 8.** Concept lattice obtained by first attributes set (Corel Image Database)

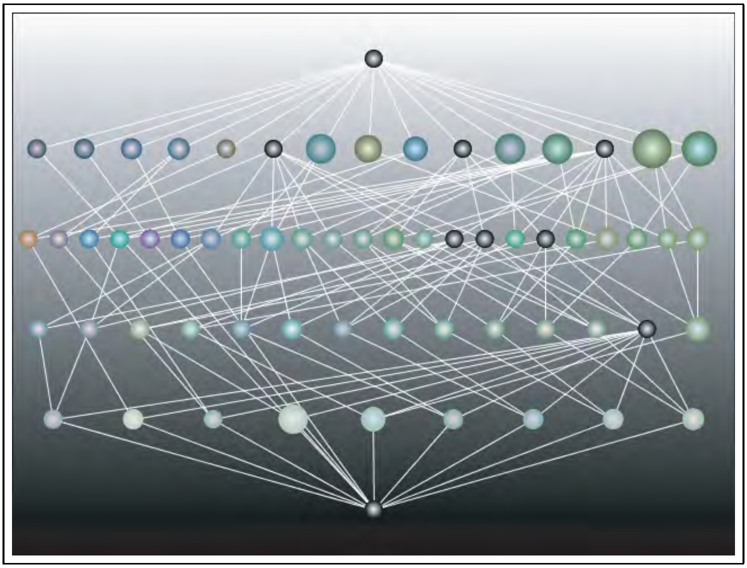


**Fig. 9.** Concept lattice obtained by second attributes set (Corel Image Database)

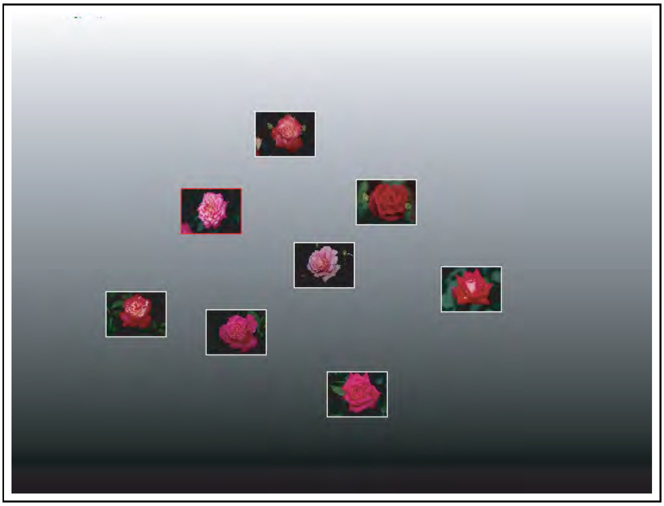


**Fig. 10.** Concept 'flower', attributes : Hue = Red, Saturation = Low

uses two fixed attribute sets, this means that the proposed system generates two attribute sets by the interaction function. The first set is composed of 15 attributes (12 color quantized regions, and 3 brightness levels). The second set

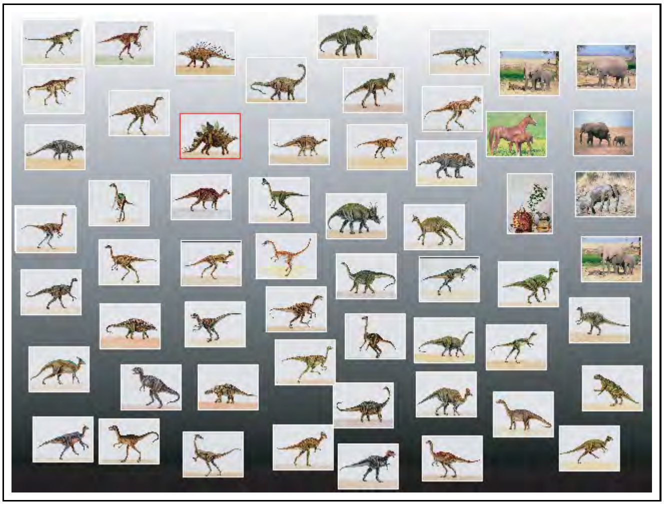


**Fig. 11.** Concept 'dinosaur', the attributes : Saturation = High

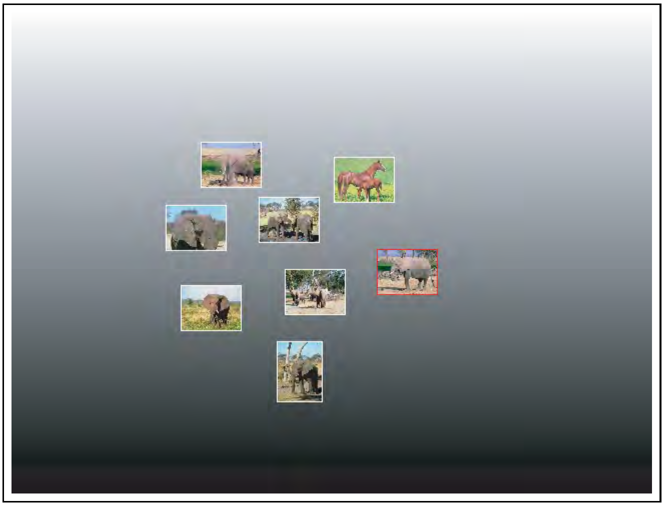


**Fig. 12.** Concept 'elephant', RGB color space = Red and Green

of attributes is composed of 24 attributes (20 color quantized regions, and 4 brightness levels).

The concept lattices obtained are shown in Fig. 8 (first attribute set) and Fig. 9 (second attribute set), respectively. In these figures, the size of an element is proportional to the number of images which belong to each concept. The color of an element of the lattice represents the average color of the images it contains.

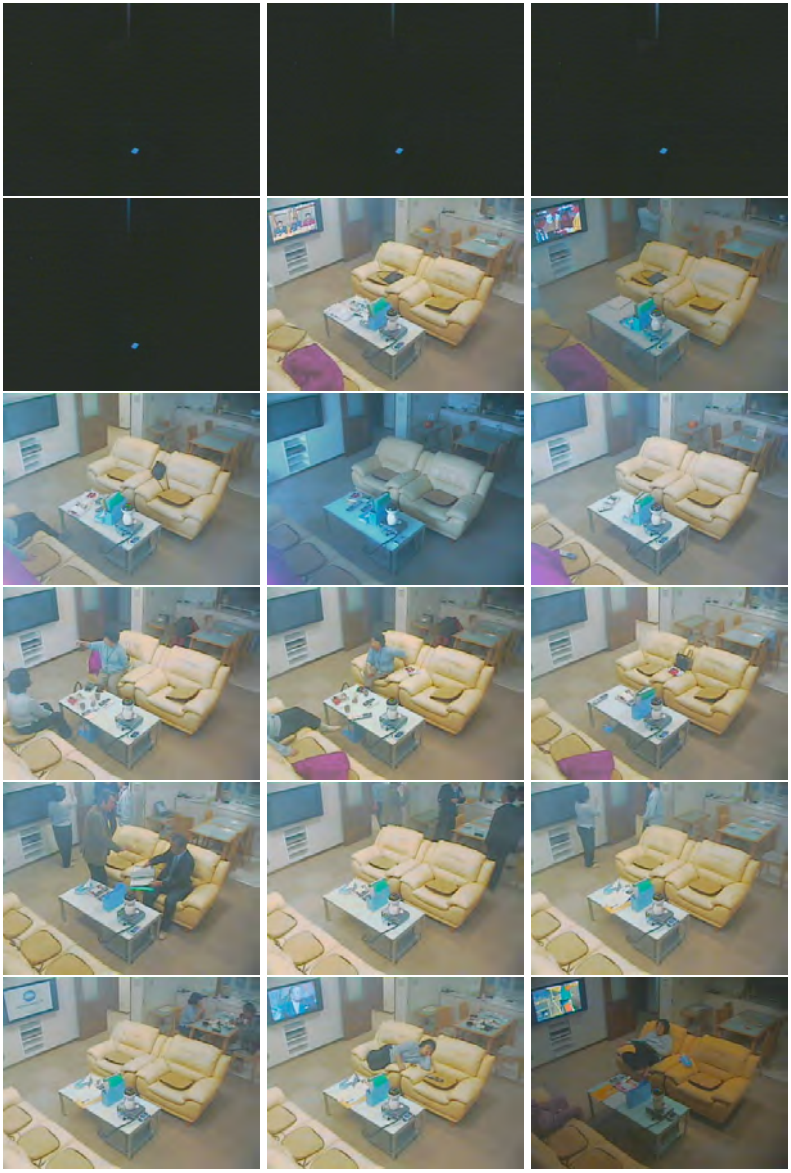


**Fig. 13.** Example Images, Ubiquitous Home Database

Therefore, using the proposed visualization platform of the concept lattice, we can recognize a perspective structure of the whole image database.

The proposed system can focus on each of the elements of the concept lattice. The user can zoom in at the level of any element and he can view the images included in each concept as shown in Figs. 10, 11, and 12. These can

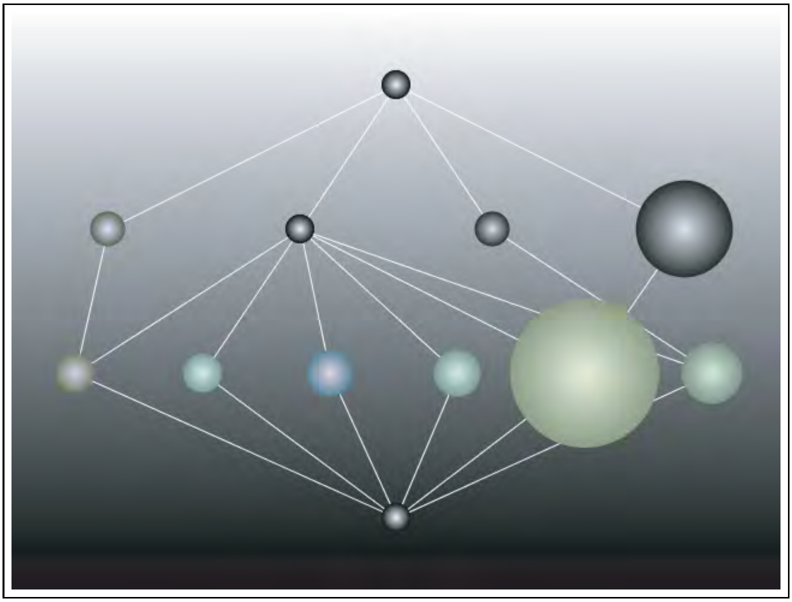


**Fig. 14.** Concept lattice obtained by first attributes set (Ubiquitous Home Database)

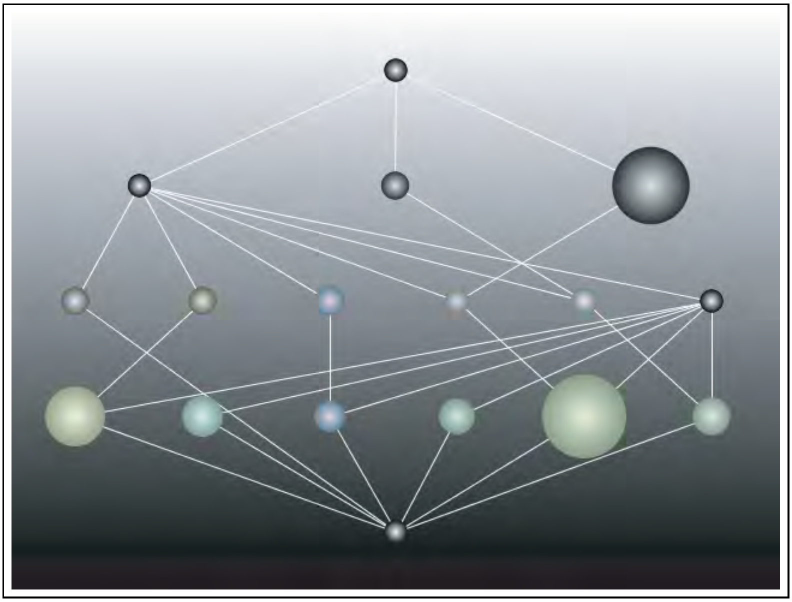


**Fig. 15.** Concept lattice obtained by second attributes set (Ubiquitous Home Database)

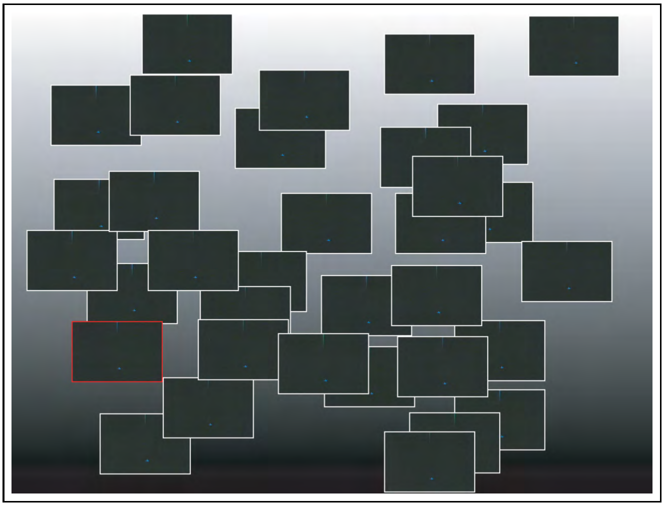


**Fig. 16.** Obtained scene 1 by proposed system

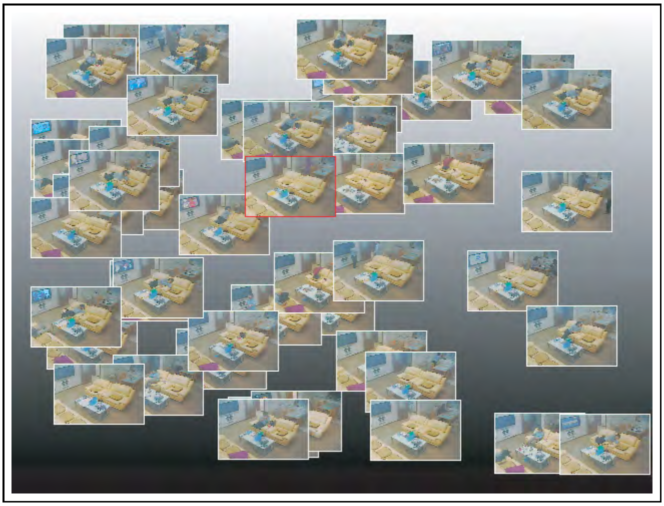


**Fig. 17.** Obtained scene 2 by proposed system

be recognized as belonging to the ‘flower’, ‘dinosaur’ or ‘elephant’ categories, respectively. Of course, the attributes of these concepts express the features in the ‘flower’, ‘dinosaur’ and ‘elephant’ concepts.

By visualization based on formal concept analysis, we found the following knowledge with respect to Corel Image Gallery:

- The ‘flower’ concept is composed of the attributes : Hue = Red, Saturation = Low
- The ‘Dinosaur’ concept is composed of the attributes : Saturation = High
- The ‘elephant’ concept has the following attributes : RGB color space = Red and Green

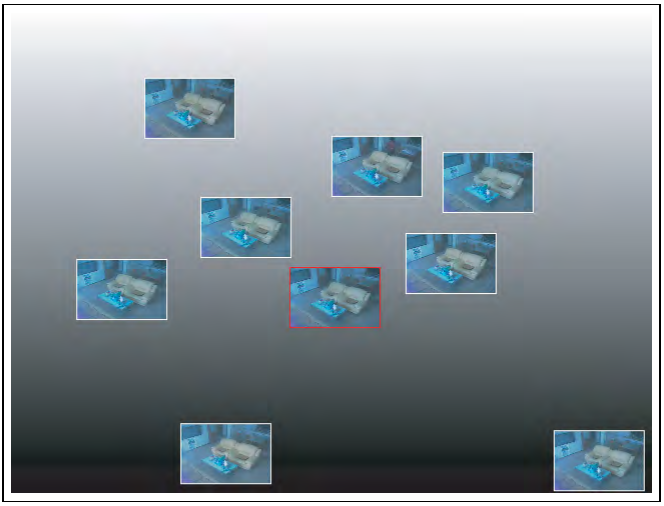


**Fig. 18.** Obtained scene 3 by proposed system

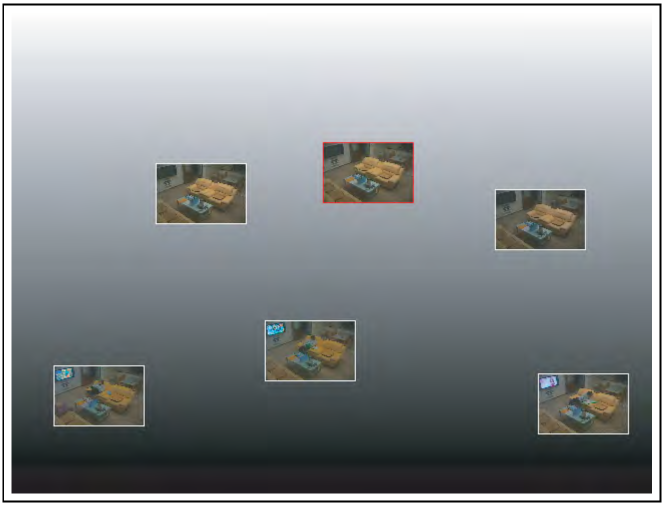


**Fig. 19.** Obtained scene 4 by proposed system

### 4.2 Ubiquitous Home Database Visualization

In this subsection, we perform a lattice visualization experiment with the *ubiquitous home database* produced by NICT (National Institute of Information and Communication Technology). Some example images are shown in Fig. 13. As it can be seen from these images, they are obtained by fixed cameras. Unlike the previous section's data, images of the ubiquitous home database have almost the same picture composition. In this visualization experiment, we confirm the effectiveness of the proposed system for scene extraction and recognition. As attribute sets, we employ the same two sets as in the previous section.

The concept lattices obtained are shown in Fig. 14 (first attribute set) and Fig. 15 (second attribute set), respectively. Figures 16 - 19 show some extracted scenes. As it can be seen from these results, the proposed visualization system is useful for scene recognition.

## 5 Proposed Lattice Structure Visualization and Granular Computing Based Human Centric Information Processing

At the beginning of the manuscript we have pointed out the conceptual relationships between the proposed system and the granular computing paradigm. In the present section we compare in view of the experimental results shown, the proposed visualization system with the basic concepts of granular computing.

These results show a strong relationship between granular computing and the proposed, formal concept analysis-based lattice structure visualization system. Indeed, we can see a correspondence between our system and basic concepts of Granular Computing.

1. The proposed system's information granules correspond to the elements of the concept lattice while the partitioning corresponds to the quantization of features.
2. It is easy to see that the concept lattice is a realization of the perspective, hierarchical nature of granular computing. In the concept lattice, upper elements stand for big information granules, and by duality, lower elements represent small granules.
3. The ability of the proposed system for the derivation of knowledge was shown in the proposed examples. It is interesting to remark here that the proposed system also caries the property to be highly intuitive, and so it allows the empirical verification of the results.
4. Another notable result of this manuscript is that the proposed system has interaction functions to adjust the quantization number, and to select the attributes for obtaining a suitable concept lattice.

We can now conclude by pointing out that altogether, the features of the proposed system show the characteristics of *Human-Centric Information Processing based on Granular Computing.*

## 6 Conclusions

In order to extract understandable information with suitable sizes from huge volumes of image data, a lattice visualization system based on granular computing and formal concept analysis has been proposed. The proposed system generates the information granules as elements in the concept lattice, from the cross table composed of objects and attributes. The objects and attributes are images and features respectively. The interaction function proposed here, allows users to adjust the quantization of features and to select desired attributes. In the visualized lattice, users can easily recognize the perspective structure of a whole image database.

Since we can recognize in this approach information granules as elements of our lattice and partitioning of granular elements as the quantization of attributes, the proposed lattice visualization system is a realization of the important, granular computing paradigm.

The proposed system has been developed on an ordinal computer (CPU = 2.13GHz, MM = 2GB) based on *Processing* language, and two visualization experiments have been performed. In these experiments, we have used the Corel Image Gallery and the Ubiquitous Home Database. Through these experiments, we have confirmed that the proposed system is able to visualize huge image databases in an intuitive way, and also it becomes suitable for scene recognition from a huge amount of data.

## References

1. Bargiela, A., Pedrycz, W.: Granular Computing – An Introduction. Kluwer Academic Publishers, Dordrecht (2002)
2. Bargiela, A., Pedrycz, W.: Toward a theory of granular computing for human-centered information processing. IEEE Transactions on Fuzzy Systems 16, 320–330 (2008)
3. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Plenum Pub. Corp. (2002)
4. Boutell, M., Luo, J.: Beyond Pixels: Exploiting Camera Metadata for Photo Classification. Pattern Recognition 38, 935–946 (2005)
5. Chen, C.: Information Visualization: Beyond The Horizon. Springer, Heidelberg (2006)
6. Cooper, M., et al.: Temporal Event Clustering for Digital Photo Collections. ACM Transactions on Multimedia Computing, Communications and Applications 1(3), 269–288 (2005)
7. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order, 2nd edn. Cambridge University Press, Cambridge (2002)
8. Fry, B.: Visualizing Data. O'Reilly, Sebastopol (2008)
9. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Heidelberg (1996)
10. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Addison-Wesley, Reading (2002)
11. Kaburlasos, V.G., Ritter, G.X. (eds.): Computational Intelligence Based on Lattice Theory. Studies in Computational Intelligence, vol. 67. Springer, Heidelberg (2007)

12. Konar, A.: Computational Intelligence. Springer, Heidelberg (2005)
13. Lin, T.Y.: Granular computing on binary relations. In: Polkowski, L., Skowron, A. (eds.) Rough sets in knowledge discowery: Methodology and applications, pp. 286–318. Physica-Verlag, Heildelberg (1998)
14. Mencar, C., Fanelli, A.M.: Interpretability constraints for fuzzy information granulation. Information Sciences 178, 4585–4618 (2008)
15. Pal, S.K., Mitra, P.: Multispectral image segmentation using rough set initialized EM algorithm. IEEE Trans. Geosci. Remote Sensing 40, 2495–2501 (2002)
16. Pal, S.K., Shankar, B.U., Mitra, P.: Granular computing, rough entropy and object extraction. Pattern Recognition Letters 26, 2509–2517 (2005)
17. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets: Analysis and Design. Bradford Books (1998)
18. Reas, C., Fry, B.: Processing. MIT Press, Cambridge (2007)
19. Singh, R., Vatsa, M., Noore, A.: Integrated multilevel image fusion and match score fusion of visible and infrared face images for robust face recognition. Pattern Recognition 41, 880–893 (2008)
20. Ware, C.: Information Visualization: Perception for Design. Morgan Kaufmann, San Francisco (2004)
21. Yao, Y.Y.: Granular Computing, Computer Science. In: Proceedings of The 4th Chinese National Conference on Rough Sets and Soft Computing, vol. 31, pp. 1–5 (2004)
22. Zadeh, L.: Fuzzy sets. Information and Control 8, 338–353 (1965)
23. Zadeh, L.: Toward a theory of fuzzy granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems 90, 111–127 (1997)
24. Zadeh, L.: Is there a need for fuzzy logic? Information Sciences 178, 2751–2779 (2008)
25. http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf
26. http://research.nii.ac.jp/i-explosion/
27. http://www.view-hokkaido.jp/

# Artificial Dendritic Cells: Multi-faceted Perspectives

Julie Greensmith and Uwe Aickelin

University of Nottingham, School of Computer Science, Nottingham, NG8 1BB, UK
{jqg,uxa}@cs.nott.ac.uk

**Abstract.** Dendritic cells are the crime scene investigators of the human immune system. Their function is to correlate potentially anomalous invading entities with observed damage to the body. The detection of such invaders by dendritic cells results in the activation of the adaptive immune system, eventually leading to the removal of the invader from the host body. This mechanism has provided inspiration for the development of a novel bio-inspired algorithm, the Dendritic Cell Algorithm. This algorithm processes information at multiple levels of resolution, resulting in the creation of information granules of variable structure. In this chapter we examine the multi-faceted nature of immunology and how research in this field has shaped the function of the resulting Dendritic Cell Algorithm. A brief overview of the algorithm is given in combination with the details of the processes used for its development. The chapter is concluded with a discussion of the parallels between our understanding of the human immune system and how such knowledge influences the design of artificial immune systems.

## 1 Introduction

The human immune system (HIS) is a decentralised, robust and error tolerant system which consists of a plethora of interacting cells. This system provides protection from invading entities such as bacteria and regulates numerous bodily functions. Immunology, the study of the human immune system, encompasses multiple levels of abstraction. For the past 100 years immunology has been a reductionist science, concentrating on the precise mechanisms involved in the relationship between immune-related molecules and cells. More recently [Cohen07] immunologists are examining such components from a systemic perspective. The exact purpose of the HIS still remains elusive, though current thinking within immunology is that it provides a combination of *protection* and *regulation*. Protection involves the rapid detection of invading microorganisms termed *pathogens*, their subsequent removal from the body and the process of repair following pathogenic infection. Regulation via the immune systems involves the maintenance of a constant internal environment, namely the homeostasis of bodily processes. This includes temperature, acidity levels, growth of blood vessels, regulation of inflammatory processes and tolerance to self-cells.

As computer scientists, our interest in the immune system is as a protection system as natural parallels can be drawn between natural pathogens and threats to computer systems, such as internet based ‘viruses’ and ‘worms’[Forrest94]. To use the immune system as inspiration for computer algorithms, the construction of immune-inspired models is performed at numerous levels of abstraction, including molecular signaling

A. Bargiela, W. Pedrycz (Eds.): Human-Centric Information Processing, SCI 182, pp. 375–395.
springerlink.com

networks and models of cell. These concepts are translated into an algorithm or system through processes of abstraction and modeling.

The creation of artificial immune systems (AISs) involves the translation of basic immunological models into feasible algorithms. This requires careful modeling of immune inspired features. To achieve this successfully, it is recommended that the desired immune components are modelled at various levels of abstraction then transformed into an algorithm using a similar multi-scale ethos.The choice of functions abstracted from the natural system is heavily influenced by the methods used in experimental immunology as this limits our understanding of the immune system. Three different levels of abstraction are commonly used including the molecular level, cellular level and systemic level, with the majority of research focusing on the molecular level. Such trends within immunology influence the manner by which AISs are created with most using models of molecular interactions in terms of binding between molecules [deCastro02].

The Dendritic Cell Algorithm (DCA) is an example of an immune inspired algorithm developed using a multi-scale approach. This algorithm is based on an abstract model of *dendritic cells* (DCs). The DCA is abstracted and implemented through a process of examining and modeling various aspects of DC function, from the molecular networks present within the cell to the behaviour exhibited by a population of cells as a whole. Within the DCA information is granulated at different layers, achieved through multi-scale processing. This differs from the standard view of granular computation [Bargiela03] as such information granules do not have an explicit fuzzy component or membership function. However, their processing is performed in a similar multi-level manner and across multiple time scales forming a diverse set of information granules. Input data is in the form of two different input streams, which are combined and correlated across variable time windows. In addition such AIS algorithms are inherently human-centric in their development. They are based on a foundation of how the immune system is perceived through immunological experimentation. This ultimately forms the abstract biological model underpinning the function of immune inspired computational systems.

In this chapter we use the parallels drawn between natural DCs and the artificial population of DCs used in the DCA to illustrate the principles behind multi-scale algorithm development. The aim of this chapter is to show how such abstraction can be achieved and to stress the importance of understanding a system from multiple perspectives to produce systems that encompass several layers of information granularity. In Section two an overview of the relevant immunology is presented and the development process of multi-signal models is outlined. In addition we present an overview of the AIS algorithms developed for computer security and optimisation and draw parallels with human-centric developments in immunology. Section three introduces the DCA as a multi-resolution algorithm. Section four provides a brief description of an implemented DCA highlighting signal and antigen processing as granular computation, and Section five continues the discussion of the DCA in the context of human centric development. Finally conclusions are drawn regarding the relationships between immunology, AIS and the lessons learned from the developmental process used to create the DCA.

## 2 Background

### 2.1 Human Immune System

The human immune system (HIS) is vast, containing in excess of 10 million cells. There is no archetypal "immune cell" akin to neurones in the central nervous system. Instead the HIS is an abstract concept, a name imposed by immunologists for a collection of cells whose function is within the remit of protection and regulation. The HIS is classically subdivided into two distinct branches: the *innate* and the *adaptive* systems. The innate system is evolutionarily the oldest immune component and its role to provide a rapid response on detection of specified molecules within the body [Murphy08].

Innate cells include macrophages, natural killer cells and dendritic cells, which perform initial pathogen detection by instructing the immune system of damage and clear the surrounding tissue of any debris. Over the evolution of the species, the immune system has acquired the knowledge of which molecules indicate the presence of pathogens. Immune cells are equipped with receptors (surface bound proteins) armed to detect such molecules. These receptors are present in great number on the cells of the innate immune system. The repertoire of pathogenic recognition receptors (termed *pattern recognition receptors)* is fixed once the genome of an individual is encoded. This implies that the innate immune system cannot adapt to novel threats over the lifetime of the individual - an important task given the fact that pathogens are constantly evolving.

To keep pace within this biological arms race, the HIS also contains a population of cells that are able to dynamically restructure their receptors to adapt to new threats. Such cells of the adaptive immune system, namely B-cells and T-cells, have the ability upon cloning to reorganise the molecules of their pathogen detectors (termed *variable region receptors*) to attempt to adapt to new threats. It is the combination of the rapid response of the innate immune system, coupled with the dynamic modifications of the adaptive immune system that provides sufficient protection to ensure the survival of our species.

The current thinking of immunologists heavily influences the manner by which we construct AISs. The inspiration used as the basis of such algorithms is derived not from the immune system itself, but from human abstractions of how we believe the immune system to function. Therefore here we introduce the basic trends in immunology over the past 100 years and comment on how various human-centric streams of research in immunology has influenced the field of AIS.

In 1891, Paul Ehrlich and his colleagues [Silverstein05] postulated that the human defense mechanism against pathogens revolved around the generation of immunity through the production of antibodies. He showed that these generated antibodies are specific to the pathogen or toxin being targeted. From his perspective a paradox existed termed *horror autotoxicus* - the immune system has to ensure that invaders are controlled and deleted before an infection spreads without responding to or damaging its own cells. Following Ehrlich's work, the *clonal selection principle* was developed where the immune system is postulated to have the ability to respond to proteins - termed antigen - which do not belong to 'self' and to target antigens belonging to 'nonself i.e. pathogenic proteins.  This formed a major constituent of a theory known as central tolerance and is shown in Figure 1 as the "one-signal model".

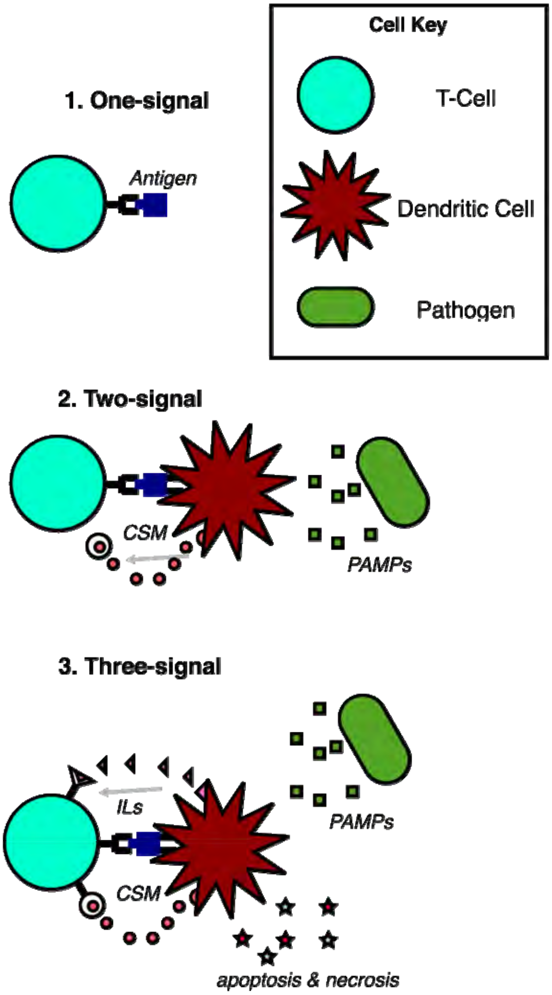


**Fig. 1.** Abstract representation of the three multiple signal models developed in immunology, including the original one signal model, the costimulation driven two signal model where the involvement of pathogens was not understood until after the model was explored, and finally the three signal model which also includes danger signals.

As the 20th century progressed, T-cells were characterised in addition to the antibody producing B-cells. In 1959, Lederberg proposed the principle of negative selection. He established the link between foetal development and the generation of tolerance to self-antigen. It is shown that in infancy, newly created T-cells are presented a sample of self-antigen. T-cells are deleted if they displaying a receptor protein which matches self-antigen with a sufficiently high binding affinity. This results in a population of T-cells acutely tuned to respond to non-self entities.

However, this response to non-self is not always an observable fact and numerous noteworthy exceptions have been discovered [Murphy07]. Four main problems have arisen questioning the credibility of central tolerance and ‘self-nonself' as the central dogma of immunology.

- Vaccinations and immunisations require adjuvants (bacterial detritus) despite the vaccination containing non-self particles;
- What the body classes as self changes over time for example in pregnancy;
- Our guts are host to colonies of bacteria which serve a symbiotic function forming the gut flora, without which we are prone to severe intestinal infections and inflammation;
- The immune system can behave inappropriately and attacks its host in the form of autoimmune diseases such as multiple sclerosis, rheumatoid arthritis and inflammatory bowel disorders, in addition to the generation of allergy to benign particles such as pollen.

The first major modification to the classical one-signal model is the addition of a secondary pathway for the activation of adaptive immune cells. This is termed *costimulation*and has been shown as a requirement for the full activation of T-cells, forming the two-signal model shown in Figure 1. Even if an antigen and T-cell bind sufficiently well, a costimulation signal is required in order for the activation of the T-cell to effector function. In order to bind to antigen, a T-cell must be ‘presented’ the antigen by a cell of the innate immune system, known as an *antigen presenting cell (APC)* such as DCs.

It is thought that for a T-cell to become activated it must be first presented its antigen by an APC in conjunction with molecules termed co-stimulatory molecules (CSM). Initially it was undetermined as to what causes APCs to express such molecules. Janeway [Janeway89] postulated that APCs produced CSMs in response to the detection of bacterial sugars, known as PAMPs - *pathogen associated molecular patterns.* These molecules are exclusively produced by pathogens as the name suggests and hence act as a signature of bacterial presence in the body. This is a ‘two-signal model’ (Figure 1) as the T-cell is given two signals, CSM and antigen.

This theory explains the need to add bacterial detritus to immunisations, and also the lack of response to changing self-proteins, as they do not have PAMPS. However, this theory alone cannot explain the lack of response by the immune system to the ‘friendly’ bacteria in the gut or the phenomenon of auto-immunity to which no pathogens are present.

One of the most recent models is the “danger theory” which incorporates a third signal. Matzinger [Matzinger94] proposed that in addition to the requirement for antigen and CSMs, T-cells also require a particular type of *interleukin,* a messenger molecule, from the APC to promote full T-cell activation. The danger theory postulates that this particular interleukin is produced by the APC in response to exposure to tissue damage. This ‘three-signal model’ is shown in Figure 1.

It is thought that the presence of something like a bacterial colony in an otherwise healthy piece of tissue will cause the tissue cells to die unexpectedly. As a result of such cell death, termed *necrosis,* the inner constituents of the tissue cells are subject to a rather chaotic degradation process. DCs in particular are shown to increase production of the relevant interleukin, IL-12, upon receipt of such indicators of cell damage.

Conversely, cells can die as part of a normal regulatory process, termed *apoptosis.* DCs exposed to the signals of apoptosis themselves produce a different kind of signal, termed IL-10. Instead of activating the T-cell, production IL-10 by DCs causes T-cell deactivation. Through DCs producing varying amounts of IL-12 versus IL-10, the T-cell is given final confirmation whether to respond to the presented antigen or to become tolerant to its presence.

Research continues in immunology to find further plausible mechanisms of immune activation. Recently, a new type of T-cell, a Th17 cell has come to the fore using a fourth signal expressed by DCs. The mechanism of action still remains unclear, though it appears that this cell is stimulated without the third interleukin signal and in the presence of a fourth signal known as TGF-β. These discoveries show that no matter what the current state of the art, such models will be continually updated and improved as we develop increasingly sophisticated techniques for the study of the function of the HIS, leading in perpetual development of AIS based on these new discoveries.

To summarise, multiple-signal models of T-cell activation have dominated much of immunology for the past century. This basic model has been subject to much debate and numerous additions incorporating different molecular activating and suppressing signals in addition to the binding of T-cell to antigen. Understanding the basics of immunology is the initial step in creating AIS algorithms. In the next section we discuss how AISs have developed in a similar manner to the multiple signal models presented in this section.

### 2.2 Artificial Immune Systems (AISs)

AISs are computer systems and algorithms inspired by the function of the HIS. There are numerous parallels in the pathway of development of AISs. As with immunology, AIS also began by using the self-nonself principles of negative and clonal selection to create the Negative Selection Algorithm, which was used primarily for applications within computer security. Clonal selection is used in a variety of immune algorithms including AIRS, which has proven to be a competitive multi-class classification system.

In comparison with other bio-inspired computing paradigms, AISs are relatively young. Forrest *et al.* first implemented negative selection in 1994 [Forrest94], based on the T-cell centric one-signal model. Exposition and exploration of this algorithm dominated the field of AIS for the following decade. The idea behind this principle is appealing to computer security - the notion of creating a computer immune system to detect computer viruses is naturally appealing as a metaphor. This algorithm uses self-nonself principles, creating a set of randomly generated ‘detectors’ tuned via a mechanism of profiling of normal behaviour, selection of detectors which deviate from normal. This results in a detector set tuned to only respond to ‘non-self’ or anomalous strings.

While negative selection generated much interest in AIS, the algorithm itself has been shown to have a number of shortcomings. The nature by which the detectors are generated relies on the initial creation of a sufficient amount of detectors to cover the potential self-nonself feature space. Obviously, as the dimensionality or size of this feature space increases, the number of detectors required to fully cover such space increases exponentially. This has been proven both experimentally [Kim01] and theoretically [Stibor06]. In addition to such scaling problems, the algorithm also is prone to the generation of false alarms or false positives. These type1 misclassification errors are thought to arise partially due to the ‘one-shot’ style of learning, and the fact that it is difficult to accurately represent what is ‘normal’ within a single bit-string [Stibor05]. Despite numerous attempts to remedy this challenge with thorough investigations of the representation [Zhou06], this algorithm does not produce results similar in calibre to that observed by the HIS.

Consequently AIS researchers have incorporated an ever-increasing amount of underlying immune-inspiration in an attempt to improve such algorithms. For example, the incorporation of a second signal was first proposed by Hofmeyr [Hofmeyr99] and implemented by Balthrop [Balthrop05], where it was shown to reduce the rates of false positives in numerous cases. As with immunology, AIS has continued to add signals to its underlying models in much the same manner as immunologists have over the past century.

Aickelin *et al.* proposed a novel approach to the development of AIS[Aickelin03] centered in the incorporation of the danger theory to AIS. Two streams of research

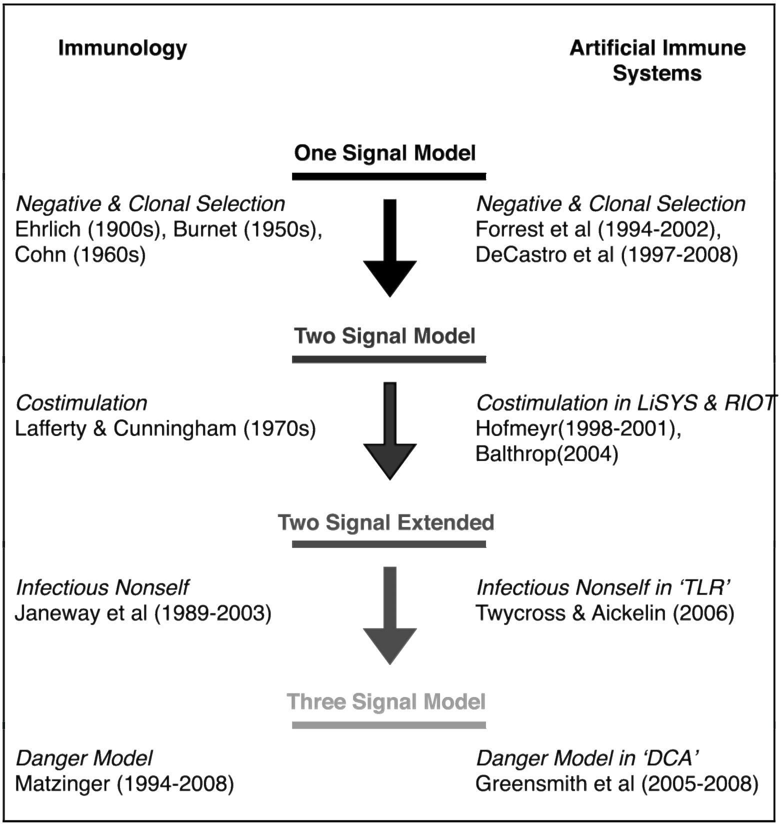

Immunology
Artificial Immune Systems
One Signal Model
Negative & Clonal Selection
Ehrlich (1900s), Burnet (1950s), Cohn (1960s)
Negative & Clonal Selection
Forrest et al (1994-2002), DeCastro et al (1997-2008)
Two Signal Model
Costimulation
Lafferty & Cunningham (1970s)
Costimulation in LiSYS & RIOT
Hofmeyr(1998-2001), Balthrop(2004)
Two Signal Extended
Infectious Nonself
Janeway et al (1989-2003)
Infectious Nonself in ‘TLR’
Twycross & Aickelin (2006)
Three Signal Model
Danger Model
Matzinger (1994-2008)
Danger Model in ‘DCA’
Greensmith et al (2005-2008)

**Fig. 2.** The parallel development of immunology and subsequently, artificial immune systems. Given the trend in artificial immune systems to work increasingly closely with immunologists, we expect that this trend will continue for the forseeable future within this field.

resulted from this proposition, one including Janeway's infectious nonself model and the other resulting in the creation of the Dendritic Cell Algorithm. Both algorithms are applied with success to the detection of network intruders, encompassing a variety of problems within such fields [Greensmith06, Twycross06].

The augmented two-cell model was implemented by Twycross and Aickelin [Twycross06] and while it was never explained as incorporating a PAMP signal (it is expressed as a 'danger signal' in their literature) it is indeed incorporating a secondary signal to a process which also requires the selection of a T-cell along with the use of an APC to provide the second signal. The second signal was derived from data out of range of characterised 'normal' data.

The developments of AIS outlined above do not focus on the development of clonal selection and idiotypic network based systems, as they do not have sufficient relevance to the development of the DCA. However, the interested reader is advised to refer to Timmis and DeCastro [Timmis02] for further details.

The development of AIS for uses within computer security in particular have inherent parallels with dogma in immunology as summarised in Figure 2. This can be attributed to the fact that AIS researchers are improving their relationships with practical immunologists as interdisciplinary collaborations become increasingly prevalent within computer science and the life sciences. This was indeed the case for the 'Danger Project' resulting in the development of the DCA. This is corollary to the fact that techniques in immunology have developed to such a level where quite detailed models can be constructed as the knowledge base expands regarding the actual function of the HIS.

## 3 Overview of the Dendritic Cell Algorithm (DCA)

In this section we give an overview of the DCA and its underlying immune inspiration. Metaphorically, DCs are the crime-scene investigators of the HIS, traversing the tissue for evidence of damage - namely the signals of apoptosis and necrosis, and for potential culprits responsible for the damage, namely antigen. More information regarding the function of natural DCs can be found in an excellent review by of the field by Lutz and Schuler [Lutz02] with a distilled version for computer scientists presented in [Greensmith07].

The DCA is derived from an abstract model of DC biology resulting in an anomaly detection algorithm that provides robust detection and correlation. Different cells process input data mapped as 'signals' acquired over different time periods. This generates individual 'snapshots' of input information that are subsequently correlated with antigens. The DCA is described in greater technical detail in numerous sources including Greensmith *et al.* [Greensmith06, Greensmith08a] and in the corresponding PhD thesis [Greensmith07].

The process of creating an algorithm such as the DCA is nontrivial, involving multiple stages of development and requires the performance of cross-disciplinary research in conjunction with immunologists. Within the framework of the Danger Project [Aickelin03], practical immunologists conducted parallel research which filled gaps in knowledge to assist in the creation of the most accurate models possible. In this section a high level description of the algorithm is provided for illustrative purposes.

The DCA is a population based algorithm, with each artificial cell acting as an agent within the system. To achieve the incorporation of our abstract model of DC function two levels of abstraction are used, namely the internal mechanisms of the cell and the overall behaviour of the cell throughout its lifetime. As an algorithm it performs filtering of input signals, correlation between signals and antigen, and classification of antigen types as normal or anomalous. Two levels are explicitly modelled, namely the internal cell procedures and the behavioural state changes.

The internal cell procedures form the lowest level of abstraction used to dictate the behaviour of the artificial DCs. This comprises the collection of antigen data and the cumulative processing of the cells input signals. Input signals are transformed into cumulative output signals acquired over time. Signal data enters the system and is stored in an array. The cell uses these signal values each time the cell update function is called. Upon acquisition of the signal values each cell performs a weighted sum equation to combine the inputs three times to produce interim output values. These interim values are added to a final output value resulting in each cell producing three 'running total' output signals. Each input signal has a weight associated to transform the input values into the three interim values. The model of this process is represented in Figure 3.

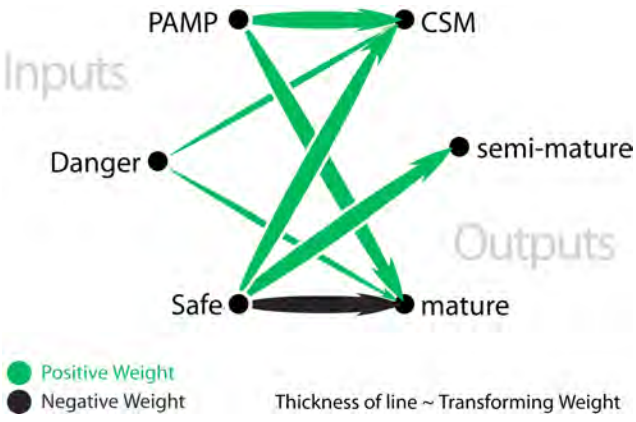


**Fig. 3.** Graphical representation of the signal processing used within each cell of the DCA. Each input signal per category is transformed to one of the three outputs. The weights used in this calculation are derived from a ratio discovered by our associate immunologists.

Each output signal is assessed each iteration. Three output signals are generated termed the costimulation value; the semi-mature output; and the mature output and their respective functions described in Table 1. The costimulation value is used to limit the lifespan of each individual cell within the DC population. Upon initialisation, each cell is assigned a threshold value, representing the lifespan of the cell. The

**Table 1.** Cumulative output signal functions for the three output signals of an artificial DC

| Output signal | Function |
|---|---|
| Costimulatory signal | Assessed against a threshold to limit the duration of DC signal and antigen sampling, based on a migration threshold |
| Semi-mature signal | Terminal state to semi-mature if greater than resultant mature signal value |
| Mature signal | Terminal state to mature if greater than resultant semimature signal value |

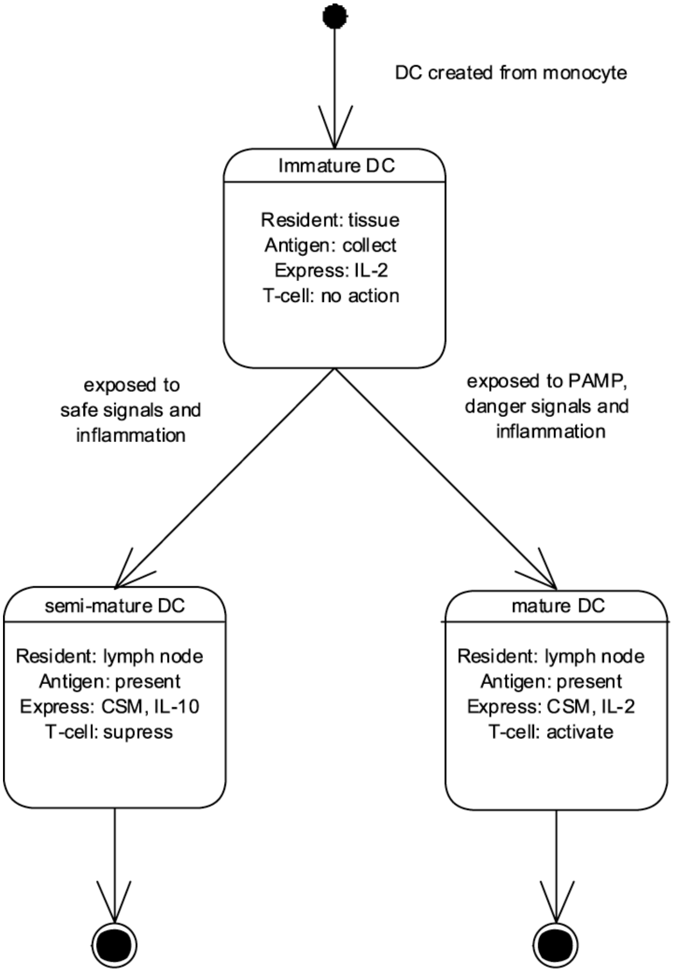


**Fig. 4.** UML state chart representing the abstract model of an individual DC

cumulative costimulation value is assessed against the lifespan threshold each iteration. Once this threshold is exceeded, the cell is removed from the population, analysed and eliminated. Upon analysis, the remaining two values are assessed.

The behavioural component is summarised in Figure 4. This level of abstraction is used to define the state changes that appear to be so pivotal to the role of the DC in the HIS. In nature DCs change state to either mature or semi-mature at a certain point. In our abstract model the DCs have perceived sufficient information when they produce a particular receptor attracting the cell to the lymph node compartment.

The costimulatory value controls the initial state change from immature to either the semi-mature or mature state. The final state is determined by the greater of the two remaining values. If the value of the semi-mature output is greater then the cell is deemed semi mature, and the same process applies should the mature signal be greater.

Each time input signals are received an antigen may also be collected (unless no antigen are generated at that timepoint). All antigen collected by the cell over its lifetime are ‘presented’ in conjunction with this context value at this analysis stage. The antigen-plus-context is used to assess the anomalous nature of such antigen. Antigen are not modified in any way by the DC, but are collected from the antigen vector and stored until presentation. The manner by which the antigen are stored has varied between the various versions of the algorithm, though this is not thought to affect the resulting performance of the algorithm.

A minimum number of ten cells are required to perform processing [Greensmith08]. The multiplicity of cells means that the algorithm uses a consensus decision generated across the population to make decisions. The output of the algorithm is an anomaly score for each antigen type, to which a threshold can apply to give a definite class prediction. Due to the time-sensitive nature of the algorithm, it is not particularly suited to randomly ordered data but is shown to have useful and robust properties when applied to challenging real-time applications [Greensmith07]. The abstract principles outlined in this section are further elaborated upon in Section 4 to demonstrate how this algorithm works in practice.

## 4 Implementing the DCA

In Section 3 a high level overview of the DCA is given. In this section a more detailed algorithmic description is given. The purpose of a DC algorithm is to correlate disparate data-streams in the form of antigen and signals and to label groups of identical antigen as ‘normal’ or ‘anomalous’. The DCA is not a classification algorithm, but shares properties with certain filtering and sorting techniques. This is achieved through the generation of an anomaly coefficient value, termed the MCAV. The labeling of antigen data with a MCAV coefficient is performed through correlating a time-series of input signals with a group of antigen. The signals are pre-normalised and pre-categorised data sources based on snapshots of preliminary experimental data, which reflect the behaviour of the system being monitored.

Categorisation of the signals is basedon the four signal model based on PAMP, danger, safe and inflammation signals. The co-occurrence of antigen and high/low signal values forms the basis of categorisation for the antigen data. The primary components of a DC based algorithm are as follows:

1) Individual DCs with the capability to perform multi-signal processing
2) Antigen collection and presentation
3) Sampling behaviour and DC maturation state changes
4) A population of DCs and their interactions with signals and antigen
5) Incoming signals and antigen, with signals pre-categorised as PAMP, danger, safe or inflammation
6) Multiple antigen presentation and analysis using ‘types’ of antigen
7) Generation of anomaly coefficient for various different types of antigen

Each cell in the population acts as an agent and has a set of instructions, which are performed every *cell update* iteration. Control of the frequency and nature of cell updates is specific to the instance of the algorithm’s implementation as is the rate of signal sampling and the number of antigen collected per iteration. Diversity is generated within the DC population by initiation of migration of the DCs at different time points i.e. the cessation of data sampling. This creates a variable time window effect throughout the DC population, which adds robustness to the system, and segments signals and antigen into variable information granules which underpin the functioning of the algorithm.

Each time a cell is updated the input signals are processed and added to the cell’s internal values to form a set of cumulatively updated output signals in addition to the collection of antigen data items. The DCs are assigned one of three states at any point in time, namely immature, semi-mature or mature. Initially the cells are all assigned the ‘immature’ state label. Upon the receipt of sufficient signal values to initiate a process termed maturation, the cell can transform to either the semi-mature or the mature state. The differences in the semi-mature and mature state are controlled by a single variable, determined by the relative differences between two output signals produced by the DCs. If, over its lifespan, the cell accumulates predominantly safe signals, the cell is classed as semi-mature, otherwise it is assigned the mature status. Whilst in the immature state, the DC has three functions, which are performed each time a single DC is updated:

1) *Sample antigen*: the DC collects antigen from an external source and places the antigen in its own antigen storage data structure.
2) *Update input signals*: the DC collects values of all input signals present in the signal storage area
3) *Calculate interim output signals*: at each iteration each DC calculates three temporary output signal values from the received input signals, with the output values then added to form the cell’s cumulative output signals.

Signal processing is performed via a weighted sum equation, bypassing the modelling of any biologically realistic gene regulatory network. A simple weighted sum equation is used in order to reduce any additional computational overheads, as the primary purpose of this algorithm is to perform anomaly detection in near to real-time. The crucial component of this procedure is the ability of the user to map normalised input data to one of the four categories of input signal (PAMP, danger, safe and inflammation). The general form of the signal processing equation is:

$$O_p = (P_w \Sigma_i P_i + D_w \Sigma_i D_i + S_w \Sigma_i S_i)\,(1+I) \qquad \forall p\ ,$$

where$P_w$, $D_w$and $S_w$are assigned weights,$P_i$, $D_i$and $S_i$are the input signal values of category PAMP (P), danger (D) or safe(S) for all signals (*i*) of that category for all output signals *p*, assuming that there are multiple signal sources per category. In this equation, the term *I* represents the inflammation signal. This sumis repeated three times, once per output signal, which are then added to the cumulative output signals. Suggested ratios for the weights are given in Table 2 where input signals are represented as *j* per category and outputs as *p* per value. Each weight can be derived from two weights directly assigned to the PAMP signals (*W1* and *W2*). The actual values used for the weights can be user defined, though the relative values determined from biological experimentation are kept constant.

**Table 2.** Derivation and interrelationship between weights in the signal processing equation, where the values of the PAMP weights are used to create the all other weights relative to the PAMP weight. *W1* is the the weight to transform the PAMP signal to the CSM output signal and *W2* is the weight to transform the PAMP signal to the mature output signal.

| | *i*= 1, PAMP | *i*= 2, Danger | *i*= 3, Safe |
|---|---|---|---|
| *p* = 1, costimulation | *W1* | *W1* / 2 | *W1* * 1.5 |
| *p* = 2, semi-mature | 0 | 0 | 1 |
| *p* = 3, mature | *W2* | *W2* / 2 | *W2* * -1.5 |

Each member of the DC population is assigned a context upon its state transition from immature to a matured state of either mature (context = 1) or semi-mature (context =0). Diversity and feedback in the DC population is maintained through the use of variable migration thresholds. The natural mechanism of DC migration is complex and not particularly well understood, involving the under and over production of numerous interacting molecules. Therefore we use a simple approximation of a thresholding mechanism using migration thresholds to assess if a DC has received sufficient information to present suitable context information along side the antigen collected during this sampling period.

Each DC in the population is assigned a “migration threshold value upon its creation. Following the update of the cumulative output signals, a DC compares the costimulatory signal value (CSM) with its assigned migration threshold value. If CSM exceeds the migration threshold, the cell ceases sampling input data and the resultant values and collected antigen are ‘presented’ for analysis. At this point the cell is reset (all internal values set to zero and antigen expunged) and returned to the sampling pool of cells.

The range of the migration thresholds assigned throughout the population is also a user-defined parameter. Previously random, Gaussian and uniform distributions have been used to provide the population with this diversity with respect to the range. We have used simple heuristics to define the limits of such ranges of threshold value, relating to the median values of the input signal data, and as a result are data-specific. The net result of this is that different members of the DC population ‘experience’ different sets of signals across a time window. If the input signals are kept constant, this implies that members of the population with low values of migration threshold present antigen more frequently, and therefore produce a tighter couple between current signals and current antigen. Conversely, DCs with a larger migration threshold may sample for a longer duration, producing relaxed coupling between potentially collected signal and context. This diversity ensures that the same information is processed in slightly different manners, resulting in noise tolerance to variation and conflict in the input data streams.

Once all data is processed or a specified number of antigen are presented (if the dataset is sufficiently large) the antigen and cell context values are collated to form the anomaly scores for each antigen type. The antigen data used with the DCA are an enumerated type variable, with multiple antigen of the same value forming a single antigen type. For example, a running process on a CPU has a process ID, and the antigen can be a representation of the process ID generated each time the process invokes a system call.

Antigens are collected by different DCsthat have experienced different snapshots of signal data. Therefore to analyse an antigen type one must average the experience of the DC population for that particular type. The value we assign per antigen type is termed the *mature context antigen value* or *MCAV*. This is a real value between zero and one: the closer this value is to one the greater the probability that this antigen type is anomalous. The MCAV is the sum of the number of individual antigen presented in the mature context divided by the total number of antigen presented for a single antigen type. This forms an average context value for each antigen type calculated from information derived using the population dynamics of the algorithm. The creation of this value also adds robustness as it cancels out any errors made by individuals in the DC population. At the core of this algorithm is a combination of numerical signal data processed at the lowest level of granularity, correlated with enumerated type antigen at a higher level of abstraction, which when brought together results in a robust anomaly detection paradigm. A generic version of the DCA is shown in Figure 5.

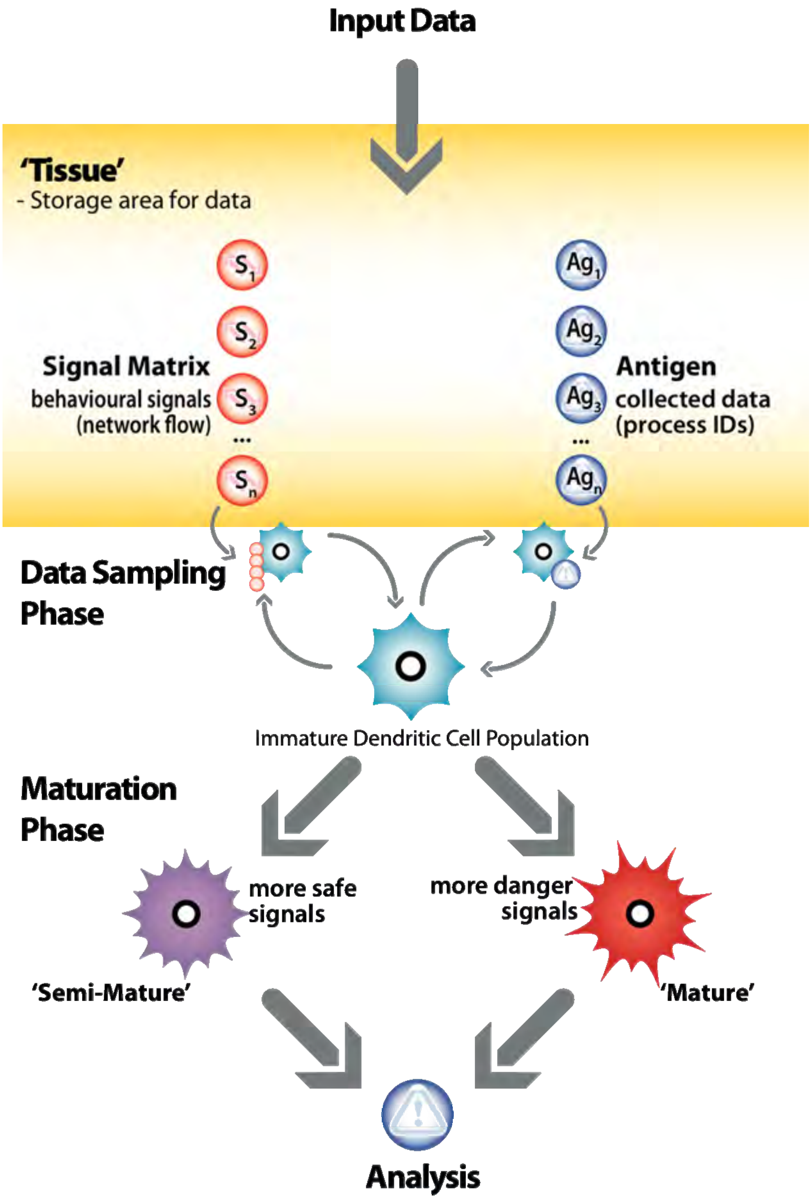


**Fig. 5.** A high level overview of the DCA as a system, with data flowing in to the signal matrix and antigen storage areas, and antigen types presented for analysis where the MCAV anomaly values are generated.

The DCA has been applied to numerous anomaly detection problems where signal and antigen mapping is possible. Such scenarios include the detection of port scans and internal intrusions. The detection of internal intrusions with the DCA formed a significant development for this algorithm, with extensive experimentation and analysis performed [Greensmith08a]. The dataset used was derived from monitoring a real host machine under a variety of experimental scenarios, such as emulating busy 'mid-morning' periods and performing scanning attacks under different network conditions. The objective of the insider experiments is to assess the DCA's performance when applied to the detection of slow and stealthy port scans. The antigen types are captured process IDs generated by the host machine each time a monitored process invokes a system call. The seven used signals are monitored from various system attributes of the monitored host:

1) PAMP1 : Number of ICMP destination unreachable errors received per second;
2) PAMP2 : Number of TCP Reset packets received per second;
3) Danger1 : Sending of network packets per second;
4) Danger2 : Ratio of TCP to all other packets per second;
5) Safe1 : Rate of change of sending network packets per second;
6) Safe2 : Average TCP packet size;
7) Inflammation: Presence of a remote root login.

In these experiments we show that the DCA has the ability to discriminate between the standard running processes on a monitored machine and an anomalous sustained port scan, performed by an emulated internal intruder. The results of this study also highlight a susceptibility of the algorithm to the 'bystander' effect, as a small number of false positives are generated to a normal process if it is equally as active as an anomalous process at exactly the same time. For this study the DCA is compared against a neural network based Self Organising Map (SOM) approach. Significant statistical differences were found in the performance of the two algorithms, with further one-sided nonparametric statistical tests concluding that the performance of the DCA is superior to that of a standard SOM, when comparing antigen type segment sizes of 10000. For full experimental details and a comprehensive analysis of this comparative study refer to Greensmith *et al.* [Greensmith08a], and [Gu08] for a comparative study of the DCA, negative selection and other machine learning algorithms.

## 5 DCA Development

Numerous stages are involved in creating an immune inspired algorithm. Following the description given in the previous section, here the process is presented by which this algorithm was designed and implemented. This process consisted of numerous stages and commenced by examining the interactions between DCs and T-cells. Once this information was compiled it became apparent that DCs perform a crucial role in mediating between the innate and adaptive immune system.

DCs appear to be a key cell in the immunological decision making process. The model generated at this stage involved multiple signal processing pathways within the DC itself in addition to complex interactions with a variety of adaptive immune cells. This model is highly complicated and is not suitable for direct transformation into an algorithm as it contained too many interactions. An abstract model of this process was required and developed.

The core of the abstract model is shown in Figure 4. This model dictates the cell behaviour and groups multiple cell inputs and outputs into four categories of input signal and three categories of output signal. In this model the state changes of an individual cell are also defined. While a DC is in its signal and antigen collection phase, the cell is termed *immature*. Upon receipt of input signals (PAMPs, danger signals from necrosis and safe signals from apoptosis) the immature cell undergoes a state change to either the *mature* or *semi-mature* state.

Antigen presented by a mature cell are potentially anomalous, and antigen presented by a semi-mature cell are potentially normal. For each type of antigen the number of semi-mature versus mature presentations are counted. This metric is used to derive an anomaly score for that particular type of antigen, upon which a threshold of anomaly is applied. Antigen with a score above this threshold are classed as anomalous. This calculation allows us to dispense with the computationally intensive process of generating T-cells, but provides a similar output functionally.

This abstract model could then be taken and transformed into a feasible algorithm as outlined in Section 3.2. As shown in Figure 6, three incarnations of the algorithm have been developed, implemented and tested on a variety of applications. The initial prototype system provided a 'proof of concept', and resulted in a feasible algorithm. This system used the minimum components, using three input signals derived from the dataset, and for each data item, ten artificial DCs were used to sample both antigen (the data ID) and signals (attributes).

As a rudimentary test, the prototype DCA is applied to the Wisconsin Breast Cancer dataset, where it achieved high rates of true positives and very low rates of false positives. This investigation highlights the suitability of the algorithm for applications which require ordered input data, such as real-time anomaly detection. We demonstrated that the DCA was not suitable for solving standard machine learning problems but could be applied to problems involving intensive processing in a real-time environment.

Following the prototype, the algorithm was scaled up to become a fully working, real-time intrusion detection system [Greensmith06]. An agent-based framework, *libtissue* [Twycross06] was used as a development platform with each DC acting as an independent agent. Antigen are fed into a storage area to be randomly selected at any point by any DC. Signals are fed into a signal matrix, with each member of the DC population updated with new input information each time the matrix is updated. The mechanism used by individual DCs to produce three output signals from this input is explained in the next section. Once all antigen are fed into the system anomaly scores are calculated for each antigen type. To test this system, the DCA is initially applied to the detection of scanning activity from a monitored client machine. As with the proof of concept experiment it is shown that high rates of false positives and low rates of true positives are generated. The initial investigation was then scaled up to encompass more sophisticated scans, where the performance was similarly good. Upon comparison with a Self-Organising Map [Greensmith08a], it is shown that the DCA produces significantly fewer false positives than this established technique.

This particular version of the DCA has also been applied to the detection of a novel threat on the internet, botnets [AlHammadi08], where the DCA produced high rates of true positives and low rates of false positives in comparison to a statistical technique. Outside of computer security Kim *et al.* [Kim06] have successfully applied the DCA

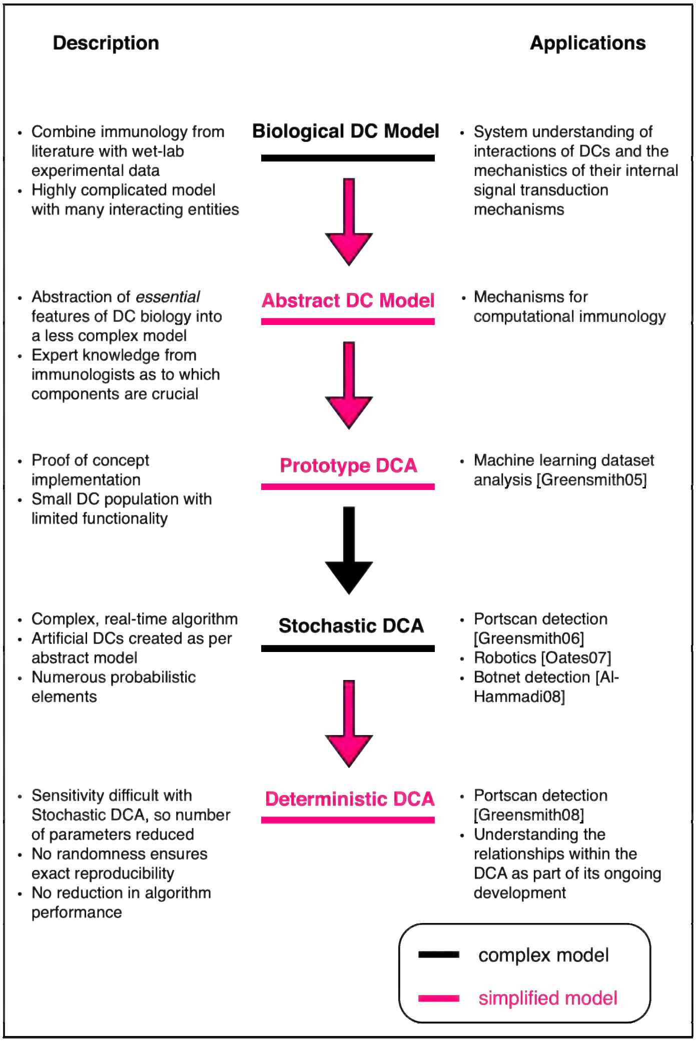

Description
Applications
Combine immunology from literature with wet-lab experimental data
Highly complicated model with many interacting entities
Biological DC Model
System understanding of interactions of DCs and the mechanistics of their internal signal transduction mechanisms
Abstraction of *essential* features of DC biology into a less complex model
Expert knowledge from immunologists as to which components are crucial
Abstract DC Model
Mechanisms for computational immunology
Proof of concept implementation
Small DC population with limited functionality
Prototype DCA
Machine learning dataset analysis [Greensmith05]
Complex, real-time algorithm
Artificial DCs created as per abstract model
Numerous probabilistic elements
Stochastic DCA
Portscan detection [Greensmith06]
Robotics [Oates07]
Botnet detection [Al-Hammadi08]
Sensitivity difficult with Stochastic DCA, so number of parameters reduced
No randomness ensures exact reproducibility
No reduction in algorithm performance
Deterministic DCA
Portscan detection [Greensmith08]
Understanding the relationships within the DCA as part of its ongoing development
complex model
simplified model

**Fig. 6.** Diagram of the DCA development process. As shown in the legend, this process cycles between complex and simple models. The appropriate level of complexity is dependent upon the use of the model, shown in the right hand column.

to the detection of misbehaviour in wireless sensor networks, where again the algorithm showed much promise. The DCA is also showing promise in the area of robotic security as demonstrated by Oates *et al.* [Oates07]. A proof of concept experiment is performed to demonstrate that the DCA could be used for basic object discrimination in a controlled environment. The same researchers have now extended this research into the theoretical domain [Oates08] through frequency tuning analysis. This research has highlighted that the DCA exhibits filter properties and moreover suggests the importance of the lifespan limit.

We had developed a seemingly successful algorithm capable of good performance across a range of problems and domains. However, this system consisted of over 15 tunable parameters, such as the number of cells, the threshold for maturation, the number of input signals, the weights for the processing of the input signals and numerous other parameters [Greensmith06]. Basic sensitivity analyses could be performed, but was difficult as due to large amounts of probabilistic elements it was not clear which components were performing which function and what exactly was performing the anomaly detection. We suspect that the key to the algorithm is the time-sensitive correlation between processed signals and collected antigen combined with a consensus decision taken across a population of cells. Due to the sheer amount of factors and parameters it was not obvious how we could analyse such a system to the degree of accuracy required.

The most recent incarnation of the DCA is a deterministic version. This remedies the problem of excessive stochastic elements and assists in proving to our community and the bio-inspired community at large not only that the algorithm can produce good results. In addition, we understand why it produces such results. Investigations of the time delay between signals and antigen have assisted in improving our understanding of how the correlation between these two sources of data is performed [Greensmith08].

In addition an improved anomaly assessment and comparable results with our previous systems, it has provided a platform in which we can track individual cells and antigen through the system over numerous repetitions and achieve identical scenarios within our antigen and signal processing. This reproducibility has let us examine the various features in isolation. We aim to extend this work across a multitude of applications and to use it to perform more theoretical analyses of the algorithm. This includes discovering in which situations it is unsuitable along with finding successful applications, allowing for a fuller characterisation of the capabilities of the technique. We intend to use this system as our testbed for adding novel components to the algorithm as the state-of-the-art in immunology progresses including such components as the Th17 cells mentioned previously.

## 6 Conclusions

In this book chapter both human centric and multi-faceted development paradigms have been presented. We have shown the parallels which exist between immunology and artificial immune systems. Such parallels are in terms of development, where immunological discovery has ultimately shaped the way in which we view the immune system in order to construct immune-inspired algorithms. This phenomena may be at least partially attributed to the fact that what is of interest to immunologists is ultimately published and such resources form the basis of inspiration. Perhaps the link between immunology and AIS will become even closer as interdisciplinary collaborations within AIS become more prevalent, resulting in algorithms which actually resemble an immune system. Whether an increased amount of immunological accuracy will be of any great benefit to AIS remains to be seen. However, the close examination of immunology appears to have been fruitful for the DCA.

With the DCA two levels of abstraction were used, namely at an intra-cellular level and at a cell behaviour level. The choice to use these levels in particular was dictated by the scope of experiments performed by the collaborating immunologists. This has resulted in an algorithm which is unique as it performs filtering on input signals, correlation between signals and antigen and classification of antigen. Without such detailed immunology, the inspiration may have appeared too abstract, and the resulting system may have become over simplistic. In order to develop complex algorithms one may need to understand the complex form of the chosen system of inspiration.

Finally, it is crucial to note the manner by which the DCA was developed as shown in Figure 5. This process varied between highly complicated models and systems to simplified versions. The cycling between complex and simple is necessary - the complex models are needed in order to find the correct level of detail, with the simplification process reducing factors such as computational complexity or having to explicitly model interactions between molecules and receptors. Both types of model, simple and complex, are needed in order to find the right level of abstraction to transform an idea into a working system.

The current incarnation, the deterministic DCA, has reduced numbers of parameters and controllable elements, so the same antigen and signals are sampled by the same cell agents provided the input is kept constant. As the simple to complex cycle continues, the next step with this algorithm is to introduce stochastic elements individually. This will allow for the investigation of the algorithm behaviour in more detail, and will assist in demonstrating how much randomness is necessary in this system or similar.

**Acknowledgements.** This work is supported by the EPSRC (EP/D071976/1).

## References

[Aickelin03] Aickelin, U., Bentley, P., Cayzer, S., Kim, J., McLeod, J.: Danger Theory: The link between AIS and IDS. In: Timmis, J., Bentley, P.J., Hart, E. (eds.) ICARIS 2003. LNCS, vol. 2787, pp. 147–155. Springer, Heidelberg (2003)

[Alhammadi08] Al-Hammadi, Y., Aickelin, U., Greensmith, J.: DCA for detecting bots. In: Proc. of the Congress on Evolutionary Computation (CEC) (2008)

[Balthrop04] Balthrop, J.: RIOT: A responsive system for mitigating computer network epidemics and attacks. Master's thesis, University of New Mexico (2005)

[Bargiela03] Bargiela, A., Pedrycz, W.: Granular Computing: An Introduction. Springer International Series in Engineering and Computer Science, vol. 717 (2003)

[Cohen07] Cohen, I.R.: Real and artificial immune systems: computing the state of the body. Nature Reviews in Immunology 7(7), 569–574 (2007)

[deCastro02] de Castro, L.N., Von Zuben, F.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems 6(3), 239–251 (2002)

[Forrest94] Forrest, S., Perelson, A., Allen, L., Cherukuri, R.: Self-nonself discrimination in a computer. In: Proc. of the IEEE Symposium on Security and Privacy, pp. 202–209. IEEE Computer Society, Los Alamitos (1994)

[Greensmith06] Greensmith, J., Aickelin, U., Twycross, J.: Articulation and clarification of the Dendritic Cell Algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 404–417. Springer, Heidelberg (2006)

[Greensmith07] Greensmith, J.: The Dendritic Cell Algorithm. PhD thesis, School of Computer Science, University Of Nottingham (2007)

[Greensmith08] Greensmith, J., Aickelin, U.: The Deterministic Dendritic Cell Algorithm. In: Proc. of the 7th International Conference on Artificial Immune Systems, ICARIS (to appear, 2008)

[Greensmith08a] Greensmith, J., Aickelin, U., Feyereisl, J.: The DCA-SOMe comparison: A comparative study between two biologically-inspired algorithms. Evolutionary Intelligence: Special Issue on Artificial Immune Systems 1(2), 85–112 (2008)

[Gu08] Gu, F., Greensmith, J., Aickelin, U.: Further Exploration of the Dendritic Cell Algorithm: Antigen Multiplier and Time Windows. In: Proc. of the 7th International Conference on Artificial Immune Systems (to appear, 2008)

[Hofmeyr99] Hofmeyr, S.: An immunological model of distributed detection and its application to computer security. PhD thesis, University Of New Mexico (1999)

[Janeway89] Janeway, C.: Approaching the asymptote? Evolution and revolution in immunology. Cold Spring Harbor Symposium on Quant Biology 1, 1–13 (1989)

[Kim01] Kim, J., Bentley, P.: Evaluating negative selection in an artificial immune system for network intrusion detection. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO), pp. 1330–1337 (July 2001)

[Kim06] Kim, J., Bentley, P., Wallenta, C., Ahmed, M., Hailes, S.: Danger is ubiquitous: Detecting malicious activities in sensor networks using the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 390–403. Springer, Heidelberg (2006)

[Lutz02] Lutz, M., Schuler, G.: Immature, semi-mature and fully mature dendritic cells: which signals induce tolerance or immunity? Trends in Immunology 23(9), 991–1045 (2002)

[Matzinger94] Matzinger, P.: Tolerance, danger and the extended family. Annual Reviews in Immunology 12, 991–1045 (1994)

[Murphy08] Murphy, K., Travers, P., Walport, M.: Janeway's Immunobiology, 7th edn. Garland Science (2008)

[Oates07] Oates, R., Greensmith, J., Aickelin, U., Garibaldi, J., Kendall, G.: The application of a dendritic cell algorithm to a robotic classifier. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 204–215. Springer, Heidelberg (2007)

[Oates08] Oates, R., Kendall, G., Greensmith, J.: Frequency analysis for dendritic cell population tuning: Decimating the dendritic cell. Evolutionary Intelligence: Special Issue on Artificial Immune Systems (2008)

[Silverstein05] Silverstein, A.: Paul Ehrlich, archives and the history of immunology. Nature Immunology 6(7), 639–639 (2005)

[Stibor05] Stibor, T., Mohr, P., Timmis, J., Eckert, C.: Is negative selection appropriate for anomaly detection? In: Proc. of Genetic and Evolutionary Computation Conference (GECCO), pp. 321–328 (2005)

[Stibor06] Stibor, T., Timmis, J., Eckert, C.: On permutation masks in hamming negative selection. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 122–135. Springer, Heidelberg (2006)

[Timmis02] de Castro, L., Timmis, J.: Artificial Immune Systems: A New Computational Approach. Springer, London (2002)

[Twycross06] Twycross, J., Aickelin, U.: libtissue - implementing innate immunity. In: Proc. of the Congress on Evolutionary Computation (CEC), pp. 499–506 (2006)

[Zhou06] Zhou, J., Dasgupta, D.: Applicability issues of the real-valued negative selection algorithms. In: Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2006), pp. 111–118 (2006)

# Index

# Author Index