SPRINGER BRIEFS IN STATISTICS
JSS RESEARCH SERIES IN STATISTICS

Toshio Sakata *Editor*

Applied Matrix and Tensor Variate Data Analysis

Springer

SpringerBriefs in Statistics

JSS Research Series in Statistics

The current research of statistics in Japan has expanded in several directions in line with recent trends in academic activities in the area of statistics and statistical sciences over the globe. The core of these research activities in statistics in Japan has been the Japan Statistical Society (JSS). This society, the oldest and largest academic organization for statistics in Japan, was founded in 1931 by a handful of pioneer statisticians and economists and now has a history of about 80 years. Many distinguished scholars have been members, including the influential statistician Hirotugu Akaike, who was a past president of JSS, and the notable mathematician Kiyosi Itô, who was an earlier member of the Institute of Statistical Mathematics (ISM), which has been a closely related organization since the establishment of ISM. The society has two academic journals: the Journal of the Japan Statistical Society (English Series) and the Journal of the Japan Statistical Society (Japanese Series). The membership of JSS consists of researchers, teachers, and professional statisticians in many different fields including mathematics, statistics, engineering, medical sciences, government statistics, economics, business, psychology, education, and many other natural, biological, and social sciences.

The JSS Series of Statistics aims to publish recent results of current research activities in the areas of statistics and statistical sciences in Japan that otherwise would not be available in English; they are complementary to the two JSS academic journals, both English and Japanese. Because the scope of a research paper in academic journals inevitably has become narrowly focused and condensed in recent years, this series is intended to fill the gap between academic research activities and the form of a single academic paper.

The series will be of great interest to a wide audience of researchers, teachers, professional statisticians, and graduate students in many countries who are interested in statistics and statistical sciences, in statistical theory, and in various areas of statistical applications.

More information about this series at http://www.springer.com/series/13497

Toshio Sakata
Editor

Applied Matrix and Tensor Variate Data Analysis

Springer

Editor
Toshio Sakata
Faculty of Design
Kyushu University
Fukuoka
Japan

ISSN 2191-544X ISSN 2191-5458 (electronic)
SpringerBriefs in Statistics
ISSN 2364-0057 ISSN 2364-0065 (electronic)
JSS Research Series in Statistics
ISBN 978-4-431-55386-1 ISBN 978-4-431-55387-8 (eBook)
DOI 10.1007/978-4-431-55387-8

Library of Congress Control Number: 2015959581

Printed on acid-free paper

This Springer imprint is published by SpringerNature
The registered company is Springer Japan KK

To the late Prof. Akio Kudo, Prof. Akimichi Okuma and Prof. Takashi Yanagawa, who taught me mathematical statistics in my younger days and guided me through my research life.

Preface

Statisticians encounter many types of complex data, and among them matrix and tensor data are currently popular data forms that play a vital role in various analyses. Matrix data constitute an essential data form of multivariate analysis and tensor data are comprised of a multi-way datum with which statisticians are familiar in analyzing contingency tables. The difference between matrix and tensor data forms and contingency tables is that the entries in the tensor data are real values and that the entries in the contingency tables are integers. Furthermore, tensor data can treat complex data in more flexible ways.

In this expository book we present both foundations and applications of matrix and tensor data analysis in six chapters.

Chapter 1 deals with the principal component analysis (PCA) for 3-way tensor data, popularly referred to as 3WPCA. The author of this chapter, Prof. Adachi of Osaka University, has conducted extensive research in this field and published papers in respected journals like Psychometrika and Computational Statistics. His research emphasizes hierarchical relationships among Tucker2, Tucker3, Parafac, and the common PCA. He illustrates the theory by applying it to psychological data.

Chapter 2 discusses nonnegative matrix factorization (NMF) and its application to audio signal data. After the basics, including algorithms, of NMF are explained, remarkable applications to audio signal data are given. The author of this chapter, Dr. Kameoka, is a promising award winning young researcher who works at NTT Communication Science Laboratories and is an adjunct associate professor at the Graduate School of Information Science and Technology of the University of Tokyo.

Chapter 3 introduces an application of tensor PCA to image data. The author of this chapter, Associate Prof. Inoue of Kyushu University, is a specialist of tensor principal components analysis to image data and has written many papers related to these topics in journals like Lecture Note of Computer Sciences and others. He formulated a generalized tensor PCA (GTPCA) and derived multi-linear PCA (MPCA), robust MPCA, simultaneous low-rank approximation of tensors (SLART), and robust SLART from GTPCA. Applications for image analysis are

presented in this chapter such as outlier removal, image compression, and face recognition.

Chapter 4 focuses on sparse coding theory of image data. The author of this chapter, Prof. Murata of Waseda University, is one of the founders of Independent Components Analysis in Japan and has published books about the topic. He has developed unified treatments of several principles in matrix factorizations. He shows that the difference between PCA, independent component analysis (ICA), nonnegative matrix factorization (NMF), and sparse representations are understood as different constraints when a data matrix is decomposed into a product of two matrices. Applications to image processing are also given.

Chapter 5 centers on the application of tensor normal distribution to genetic data. The author of this chapter, Dr. Akdemir, Research Associate at Cornell University, has published a pioneering paper about tensor normal distribution in the Journal of Algebraic Statistics. He expounded the estimation of mean and covariance of array (tensor) under normal distributions with missing data. Applications to real-life data involving the estimation of genotype and environment interaction and effects on possible correlated traits are detailed.

Chapter 6 introduces one-sided tests for the mean matrix of a matrix normal distribution. The late Prof. Kudo of Kyushu University, who was a pioneer of multivariate one-sided tests in Japan, taught the authors of this chapter, Associate Prof. Iwasa of Kumamoto University and Prof. Sakata of Kyushu University. Both authors have published several papers about one-sided tests in statistical journals such as the Journal of Multivariate Analysis and the Journal of Institute of Statistical Mathematics. They developed a one-sided test for mean matrix of matrix normal distributions and constructed a class of similar test statistics that includes similar tests previously obtained by Sasabuchi, Hu, and Banerjee. These works are positioned as the starting point to one-sided tests for the mean tensor of a tensor normal distribution.

All the chapters in this book distinctly develop interesting application fields for matrix and tensor data. Finally, the authors would like to express their sincere appreciation to Prof. Akimichi Takemura for his encouragment to write this book, and to Japan Statistical Society for giving us this opportunity to publish it.

Contents

Contributors

Kohei Adachi Graduate School of Human Sciences, Osaka University, Suita, Osaka, Japan

Deniz Akdemir Department of Plant Breeding and Genetics, Cornell University, Ithaca, NY, USA

Kohei Inoue Kyushu University, Minami-ku, Fukuoka, Japan

Manabu Iwasa Kumamoto University, Kumamoto, Japan

Hirokazu Kameoka The University of Tokyo, Tokyo, Japan; Nippon Telegraph and Telephone Corporation, Atsugi, Kanagawa, Japan

Noboru Murata Faculty of Science and Engineering, Waseda University, Shinjuku-ku, Tokyo, Japan

Toshio Sakata Department of Human Science of Kyushu University, Fukuka, Japan

Chapter 1
Three-Way Principal Component Analysis with Its Applications to Psychology

Kohei Adachi

Abstract The principal component analysis (PCA) procedures modified for analyzing three-way data are generally called three-way PCA (3WPCA). Popular 3WPCA procedures are known as the names, Tucker2, Tucker3, and Parafac. We describe their models and algorithms with an emphasis on a hierarchical relationship among Tucker2, Tucker3, Parafac, and the ordinary two-way PCA. After introducing the rotation techniques for Tucker3 solutions, 3WPCA procedures are illustrated with stimuli × responses × persons data observed in psychology.

Keywords Three-way principal component analysis · Hierarchical relations · Alternating least squares algorithm · Rotation · Network representation

1.1 Principal Component Analysis Modified for Three-Way Data

Principal component analysis (PCA) is a time-honored dimension-reduction technique for a two-way data matrix. Using $\ddot{\mathbf{X}} = (x_{ij})$ for the $I \times J$ data matrix, PCA can be modeled as

$$\ddot{\mathbf{X}} = \ddot{\mathbf{A}}\ddot{\mathbf{B}}' + \ddot{\mathbf{E}} = \sum_{p=1}^{P} \ddot{\mathbf{a}}_p \ddot{\mathbf{b}}'_p + \ddot{\mathbf{E}} \text{ or } x_{ij} = \sum_{p=1}^{P} \ddot{a}_{ip}\ddot{b}_{jp} + e_{ij}, \tag{1.1}$$

with $\ddot{\mathbf{A}} = (\ddot{a}_{ip}) = [\ddot{\mathbf{a}}_1, \ldots, \ddot{\mathbf{a}}_P]\,(I \times P)$, $\ddot{\mathbf{B}} = (\ddot{b}_{jp}) = [\ddot{\mathbf{b}}_1, \ldots, \ddot{\mathbf{b}}_P]\,(J \times P)$, $\ddot{\mathbf{E}} = (e_{ij})$ containing errors, and $P \leq \min(I, J)$ (e.g., ten Berge and Kiers [29]). The PCA solution is obtained by the least squares method, i.e., minimizing the sum of squared errors $\|\ddot{\mathbf{E}}\|^2 = \|\mathbf{X} - \ddot{\mathbf{A}}\ddot{\mathbf{B}}'\|^2$ over $\ddot{\mathbf{A}}$ and $\ddot{\mathbf{B}}$, where $\|\ddot{\mathbf{E}}\|$ denotes the Frobenius norm of $\ddot{\mathbf{E}}$. The minimization is attained through the singular value decomposition (SVD) of

K. Adachi (✉)
Graduate School of Human Sciences, Osaka University, 1-2 Yamadaoka, Suita, Osaka 565-0871, Japan
e-mail: adachi@hus.osaka-u.ac.jp

T. Sakata (ed.), *Applied Matrix and Tensor Variate Data Analysis*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55387-8_1

$\ddot{\mathbf{X}}$. Although the factor analysis (FA) model can also be written in the form (1.1), but FA differs from PCA, since the inter-column covariance matrix for $\ddot{\mathbf{E}}$ are assumed to be a diagonal matrix in FA (e.g., Mulaik [21]), but PCA does not have such an assumption and is thus simply formulated as the above least squares problem.

The main subject in this chapter is the modified PCA specialized for a three-way data array

$$\dddot{\mathbf{X}} = \{x_{ijk}; i = 1, \ldots, I; j = 1, \ldots, J; k = 1, \ldots, K\}. \tag{1.2}$$

Such procedures are generally refereed to as three-way PCA (3WPCA). Its origin can be found in psychometrics, which is the area for developing methodology for analyzing data observed in psychology (Kroonenberg [19]; Tucker [33]). An example of x_{ijk} in (1.2) is the score that examinee i shows for the test item j on occasion k. Then, (1.2) is the I-examinees $\times$ J-items $\times$ K-occasions array.

One of the most important psychometricians for 3WPCA is Ledyard, R. Tucker, who presented a typical procedure of 3WPCA. After his name, the procedure is called Tucker3 (Tucker [33]), which is modeled as

$$\dddot{\mathbf{X}} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R}(\mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r)g_{pqr} + \dddot{\mathbf{E}} \text{ or } x_{ijk} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a_{ip}b_{jq}c_{kr}g_{pqr} + e_{ijk} \tag{1.3}$$

Here, e_{ijk} stands for an error, $\mathbf{A} = (a_{ip}) = [\mathbf{a}_1, \ldots, \mathbf{a}_P]$ $(I \times P)$, $\mathbf{B} = (b_{jq}) = [\mathbf{b}_1, \ldots, \mathbf{b}_Q]$ $(J \times Q)$, $\mathbf{C} = (c_{kr}) = [\mathbf{c}_1, \ldots, \mathbf{c}_R]$ $(K \times R)$, and $\mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r$ denotes the tensor product providing the array $\{a_{ip}b_{jq}c_{kr}; i = 1, \ldots, I; j = 1, \ldots, J; k = 1, \ldots, K\}$ with $P \leq I$, $Q \leq J$, and $R \leq K$. The columns of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are called components and those matrices are referred to as component matrices, while three-way array

$$\dddot{\mathbf{G}} = \{g_{pqr}; p = 1, \ldots, P; q = 1, \ldots, Q; r = 1, \ldots, R\} \tag{1.4}$$

is called a core array. The Tucker3 model (1.3) can be illustrated as in Fig. 1.1: the data cube $\dddot{\mathbf{X}}$ $(I \times J \times K)$ is underlain by a smaller cube $\dddot{\mathbf{G}}$ $(P \times Q \times R)$. It

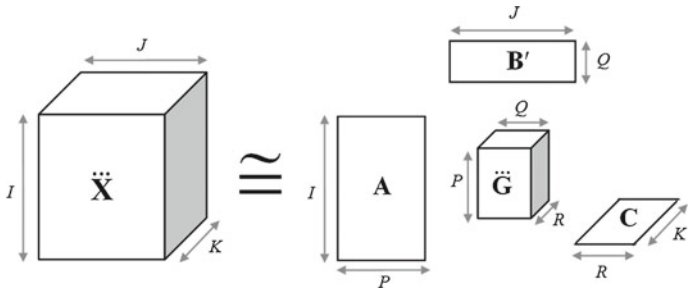


Fig. 1.1 Pictorial representation of the Tucker3 model

describes the relationships among the P, Q, and R components which correspond to the columns of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, respectively. The three matrices express the links of the components to the I, J, and K entities which define the three-modes of $\dddot{\mathbf{X}}$.

Another typical 3WPCA procedure is Parafac, which has been presented independently by Carroll and Chang [5] and Harshman [8] in psychometrics. Though the former authors called their presented procedure CANDECOMP, we use the name Parafac. Its model is a version of (1.3) in which g_{pqr} is constrained so that $g_{pqr} = 1$ for $p = q = r$ and $g_{pqr} = 0$ otherwise, with $P = Q = R$: Parafac is modeled as

$$\dddot{\mathbf{X}} = \sum_{p=1}^{P} \mathbf{a}_p \circ \mathbf{b}_p \circ \mathbf{c}_p + \dddot{\mathbf{E}} \text{ or } x_{ijk} = \sum_{p=1}^{P} a_{ip} b_{jp} c_{kp} + e_{ijk}. \tag{1.5}$$

The origin of Parafac can also be found in mathematics (Hitchcock [10]). The remaining popular 3WPCA procedure is Tucker2 (Tucker [33]), which is introduced in the next section.

In some literatures, 3WPCA is referred to as three-way factor analysis (FA). But, such naming is misleading. As described above, FA is characterized by the special assumption for error variances, but 3WPCA does not have such an assumption.

Besides psychometrics, chemometrics (for chemical data analysis) is also the area in which a number of techniques for 3WPCA have been developed. It is shown by the fact that two seminal books on 3WPCA have recently been published in chemometric field [25] and psychometric area [19], respectively.

1.2 Hierarchy in PCA and 3WPCA

Let us rearrange $\dddot{\mathbf{X}} = \{x_{ijk}\}$ in a horizontal two-way array, i.e., an $I \times KJ$ block matrix

$$\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_k, \ldots, \mathbf{X}_K]. \tag{1.6}$$

with its kth block $\mathbf{X}_k$ being the $I \times J$ matrix whose (i, j) element is x_{ijk}. For (1.6), we can apply the ordinary PCA (rather than 3WPCA) in three types of procedures. As explained in the next paragraphs, the three PCA procedures for (1.6) can be called PCA-SEP, PCA-SUP, and PCA-AVE, and they have the hierarchical relationship

$$\text{PCA-SEP} \succ \text{PCA-SUP} \succ \text{Tucker2} \succ \text{Tucker3} \succ \text{Parafac} \succ \text{PCA-AVE} \tag{1.7}$$

with popular 3WPCA procedures. Here, A $\succ$ B denotes "B is a constrained version of A". That is, 3WPCA procedures can be viewed as constrained versions of PCA rather than its extensions (Ceulemans and Kiers [6]; Kiers [14]; Murakami and Kroonenberg [22]).

PCA-SEP is the abbreviation for the PCA performed SEParately for each block $\mathbf{X}_k$ in (1.6) (Kiers and ten Berge [17]). The replacement of $\ddot{\mathbf{X}}$, $\ddot{\mathbf{A}}$, $\ddot{\mathbf{B}}$, and $\ddot{\mathbf{E}}$ in (1.1) by $\mathbf{X}_k$, $\mathbf{A}_k$, $\mathbf{B}_k'$, and $\mathbf{E}_k$ leads to the PCA-SEP model

$$\mathbf{X}_k = \mathbf{A}_k\mathbf{B}_k' + \mathbf{E}_k \tag{1.8}$$

for $k = 1, \ldots, K$. Constraining $\mathbf{A}_k$ to be invariant across k with $\mathbf{A}_k = \mathbf{A}$ in (1.8) leads to the PCA-SUP model

$$\mathbf{X}_k = \mathbf{A}\mathbf{B}_k' + \mathbf{E}_k. \tag{1.9}$$

It is rewritten as $\mathbf{X} = \mathbf{A}\mathbf{B}_{\#}' + \mathbf{E}$, i.e., the PCA model for the SUPer matrix (1.6), which gives the name PCA-SUP, with $\mathbf{B}_{\#}' = [\mathbf{B}_1', \ldots, \mathbf{B}_K']$ $(P \times KJ)$ and $\mathbf{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_K]$ $(I \times KJ)$ (Kiers [14]).

Tucker2 is the least restrictive method in 3WPCA (Tucker [33]), which is introduced by constraining $\mathbf{B}_k$ as $\mathbf{B}_k = \mathbf{B}\mathbf{H}_k'$ in PCA-SUP (1.9), where $\mathbf{H}_k$ is of order $P \times Q$. That is, Tucker2 is modeled as

$$\mathbf{X}_k = \mathbf{A}\mathbf{H}_k\mathbf{B}' + \mathbf{E}_k, \tag{1.10}$$

for $k = 1, \ldots, K$. The block matrix $\mathbf{H} = [\mathbf{H}_1, \ldots, \mathbf{H}_K]$ $(P \times KQ)$ is called an extended core array. A feature of Tucker2 is that the component matrices are invariant among k and the extended core array explains the differences in data sets $\mathbf{X}_k$ across k.

Tucker3 is a version of Tucker2 with $\mathbf{H}_k$ constrained to be the linear combination of $\mathbf{G}_r$ whose (p, q) element is g_{pqr} in (1.4):

$$\mathbf{H}_k = c_{k1}\mathbf{G}_1 + \cdots + c_{kR}\mathbf{G}_R = \sum_{r=1}^{R} c_{kr}\mathbf{G}_r \tag{1.11}$$

with $R \leq K$. The Tucker3 model is thus expressed as

$$\mathbf{X}_k = \mathbf{A}\sum_{r=1}^{R} c_{kr}\mathbf{G}_r\mathbf{B}' + \mathbf{E}_k, \tag{1.12}$$

which can be rewritten as (1.3). Further, let $\mathbf{G}_r$ in (1.12) be restricted to the matrix filled with zeros except the rth diagonal element being one with $P = Q = R$, Then, we have the Parafac model

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{B}' + \mathbf{E}_k, \tag{1.13}$$

which can be rewritten as (1.5), where $\mathbf{D}_k = \mathrm{diag}(c_{k1}, \ldots, c_{kP})$ denotes the diagonal matrix whose diagonal elements are $c_{k1}, \ldots, c_{kP}$.

The Parafac model (1.13) with $\mathbf{D}_k$ constrained to be the identity matrix $\mathbf{I}_P$ leads to

$$\mathbf{X}_k = \mathbf{A}\mathbf{B}' + \mathbf{E}_k. \tag{1.14}$$

This can be called a PCA-AVE model, as it is equivalent to the PCA model for the AVEraged data matrix $\bar{\mathbf{X}} = K^{-1}\sum_k \mathbf{X}_k$. This equivalence is shown as follows: the least squares function for (1.14) can be written as $f_1 = K^{-1}\sum_k \|\mathbf{E}_k\|^2 = K^{-1}\sum_k \|\mathbf{X}_k - \mathbf{A}\mathbf{B}'\|^2 = K^{-1}\sum_k \|\mathbf{X}_k\|^2 - 2\mathrm{tr}\bar{\mathbf{X}}'\mathbf{A}\mathbf{B}' + \|\mathbf{A}\mathbf{B}'\|^2$, while PCA for $\bar{\mathbf{X}}$ is formulated as minimizing $f_2 = \|\bar{\mathbf{X}} - \mathbf{A}\mathbf{B}'\|^2 = \|\bar{\mathbf{X}}\|^2 - 2\mathrm{tr}\bar{\mathbf{X}}'\mathbf{A}\mathbf{B}' + \|\mathbf{A}\mathbf{B}'\|^2$. That is, the terms relevant to $\mathbf{A}$ and $\mathbf{B}$ are identical between f_1 and f_2.

In (1.7), the least restrictive PCA-SEP (1.8) is found to be the best in the goodness of fit for data. But, PCA-SEP has crucial indeterminacy shown by the fact that (1.18) is rewritten as $\mathbf{X}_k = \mathbf{A}_k\mathbf{S}_k\mathbf{S}_k^{-1}\mathbf{B}_k' + \mathbf{E}_k$ with $\mathbf{S}_k$ an arbitrary nonsingular matrix: if $\mathbf{A}_k$ and $\mathbf{B}_k$ are the solutions, $\mathbf{A}_k\mathbf{S}_k$ and $\mathbf{B}_k\mathbf{S}_k^{-1\prime}$ can also be viewed as the solutions for $\mathbf{A}_k$ and $\mathbf{B}_k$. It implies that the comparison of $\mathbf{A}_k$ and $\mathbf{B}_k$ across k does not make sense without identifying $\mathbf{S}_k$ in a reasonable manner (Adachi [2, 4]). The Tucker2 and 3 models also have indeterminacy, but it is rather exploited to give interpretable solutions as described in Sect. 1.4.

1.3 Alternating Least Squares Algorithm

As in the ordinary PCA, the solutions for 3WCA are also obtained by minimizing the sum of squared errors, but they are not given explicitly. Thus, iterative procedures are used, which are included in a family of alternative least squares (ALS) algorithm (e.g., ten Berge [28]). We first describe the Parafac algorithm, which are followed by those for Tucker3 and 2.

1.3.1 Parafac Algorithm

For Parafac, some different algorithms have been presented (Carroll and Chang [5]; Harshman [8]; Kiers and Krijnen, [15]; ten Berge [28]). Among them, we describe the algorithm using the block data matrix (1.6) (Smilde et al. [25]). For the matrix, the Parafac model (1.5) or (1.13) is rewritten as

$$\mathbf{X} = \mathbf{A}(\mathbf{C} \bullet \mathbf{B})' + \mathbf{E}. \tag{1.15}$$

Here, $\mathbf{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_K]$ is an $I \times KJ$ error matrix and $\bullet$ denotes the Khatri-Rao product (Rao and Mitra [27]): $\mathbf{C} \bullet \mathbf{B} = [\mathbf{c}_1, \ldots, \mathbf{c}_P] \bullet [\mathbf{b}_1, \ldots, \mathbf{b}_P] = [\mathbf{c}_1 \otimes \mathbf{b}_1, \ldots, \mathbf{c}_P \otimes \mathbf{b}_P]$ ($KJ \times P$) with $\otimes$ denoting the Kronecker product (e.g., Harville [9]). The Parafac model (1.15) can further be rewritten as $\mathbf{X}^{\#} = \mathbf{B}(\mathbf{C} \bullet \mathbf{A})' + \mathbf{E}^{\#}$ and $\mathbf{X}^{*} = \mathbf{C}(\mathbf{B} \bullet \mathbf{A})' + \mathbf{E}^{*}$, where $\mathbf{X}^{\#}$ ($J \times KI$) and $\mathbf{X}^{*}$ ($K \times JI$) are obtained by rearranging (1.2) as

$$\mathbf{X}^{\#} = [\mathbf{X}_1', \ldots, \mathbf{X}_K'], \tag{1.16}$$

$$\mathbf{X}^{*} = [\mathrm{vec}(\mathbf{X}_1), \ldots, \mathrm{vec}(\mathbf{X}_K)]', \tag{1.17}$$

with $\mathbf{E}^{\#}$ and $\mathbf{E}^{*}$ the rearranged versions of $\mathbf{E}$, and vec() denoting the vec operator (e.g., Harville [9]). Thus, the sum of squared errors $\|\mathbf{E}\|^2$ for Parafac is expressed in three forms as in

$$f_{\mathrm{P}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{X} - \mathbf{A}(\mathbf{C} \bullet \mathbf{B})'\|^2 = \|\mathbf{X}^{\#} - \mathbf{B}(\mathbf{C} \bullet \mathbf{A})'\|^2 = \|\mathbf{X}^{*} - \mathbf{C}(\mathbf{B} \bullet \mathbf{A})'\|^2 \quad (1.18)$$

The minimization of (1.18) over $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be attained by alternately solving the following problems:

(1) $\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{A}(\mathbf{C} \bullet \mathbf{B})'\|^2$ with $\mathbf{B}$ and $\mathbf{C}$ kept fixed;
(2) $\min_{\mathbf{B}} \|\mathbf{X}^{\#} - \mathbf{B}(\mathbf{C} \bullet \mathbf{A})'\|^2$ with $\mathbf{A}$ and $\mathbf{C}$ kept fixed;
(3) $\min_{\mathbf{C}} \|\mathbf{X}^{*} - \mathbf{C}(\mathbf{B} \bullet \mathbf{A})'\|^2$ with $\mathbf{A}$ and $\mathbf{B}$ kept fixed.

Here, we can find every problem to be the regression of data onto the matrix defined by the Khatri-Rao product. For example, $\|\mathbf{X}-\mathbf{A}(\mathbf{C}\bullet\mathbf{B})'\|^2 = \|\mathbf{X}'-(\mathbf{C}\bullet\mathbf{B})\mathbf{A}'\|^2$ in (1) is the loss function for the regression of $\mathbf{X}'$ onto $(\mathbf{C} \bullet \mathbf{B})$ with $\mathbf{A}'$ a coefficient matrix. Its solution is explicitly given by $\mathbf{A}' = (\mathbf{C} \bullet \mathbf{B})^{+}\mathbf{X}'$, or equivalently, $\mathbf{A} = \mathbf{X}(\mathbf{C} \bullet \mathbf{B})'^{+}$, with $(\mathbf{C}\bullet\mathbf{B})'^{+}$ the Moore-Penrose inverse of $(\mathbf{C}\bullet\mathbf{B})'$. In parallel manners, the solutions for (2) and (3) can be obtained. Thus, the Parafac algorithm for minimizing (1.18) can be described as follows:

Step 1. Initialize $\mathbf{B}$ and $\mathbf{C}$.
Step 2. Update $\mathbf{A} = \mathbf{X}(\mathbf{C} \bullet \mathbf{B})'^{+}$
Step 3. Update $\mathbf{B} = \mathbf{X}^{\#}(\mathbf{C} \bullet \mathbf{A})'^{+}$
Step 4. Update $\mathbf{C} = \mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})'^{+}$
Step 5. Finish if convergence is reached; otherwise, back to Step 2.

A drawback of Parafac is that it sometimes provides degenerated solutions in which $\mathbf{A}$, $\mathbf{B}$, or $\mathbf{C}$ is nearly rank deficient (Smilde et al. [25]). A remedy for avoiding such solutions is to impose the column-orthonormality constraints for two of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ (Kroonenberg [19]; Smilde et al. [25]). Here, let us consider the constraints

$$\mathbf{A}'\mathbf{A} = \mathbf{I}_P, \ \mathbf{B}'\mathbf{B} = \mathbf{I}_P. \quad (1.19)$$

This version is thus formulated as minimizing (1.18) subject to (1.19). It can be attained by alternately solving the above problems (1), (2), and (3) with (1.19).

The constrained problem (1) is minimizing $f_{\mathrm{P}} = \|\mathbf{X}-\mathbf{A}(\mathbf{C}\bullet\mathbf{B})'\|^2$ over $\mathbf{A}$ under $\mathbf{A}'\mathbf{A} = \mathbf{I}_P$. Using it, f_{P} can be rewritten as $f_{\mathrm{P}} = \|\mathbf{X}\|^2 - 2\mathrm{tr}(\mathbf{C}\bullet\mathbf{B})'\mathbf{X}'\mathbf{A} + \|(\mathbf{C}\bullet\mathbf{B})\|^2$, which shows that the problem amounts to maximizing the linear form $\mathrm{tr}(\mathbf{C}\bullet\mathbf{B})'\mathbf{X}'\mathbf{A}$ over $\mathbf{A}$ subject to $\mathbf{A}'\mathbf{A} = \mathbf{I}_P$. It can be attained through the SVD defined as $\mathbf{X}(\mathbf{C}\bullet\mathbf{B}) = \mathbf{K}\Lambda\mathbf{L}'$ with $\mathbf{K}'\mathbf{K} = \mathbf{L}'\mathbf{L} = \mathbf{I}_P$ and Λ a $P \times P$ diagonal matrix: the optimal $\mathbf{A}$ is given by $\mathbf{A} = \mathbf{K}\mathbf{L}'$ (ten Berge [28]). In a parallel manner, the solution of the constrained problem (2) is given by $\mathbf{B} = \mathbf{K}^{\#}\mathbf{L}^{\#\prime}$, whose right side hand is obtained through the SVD $\mathbf{X}^{\#}(\mathbf{C} \bullet \mathbf{A}) = \mathbf{K}^{\#}\Lambda^{\#}\mathbf{L}^{\#\prime}$. Since the remaining parameter $\mathbf{C}$ is unconstrained, its solution is obtained in the same manner as in the above Step 4: for given $\mathbf{A}$ and $\mathbf{B}$ the optimal $\mathbf{C}$ is given by $\mathbf{C} = \mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})'^{+}$ which is simplified as $\mathbf{C} = \mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})$ using (1.19). Thus, the Parafac algorithm subject to (1.19) is listed as follows:

Step 1. Initialize $\mathbf{B}$ and $\mathbf{C}$.
Step 2. Update $\mathbf{A}$ with $\mathbf{A} = \mathbf{K}\mathbf{L}'$
Step 3. Update $\mathbf{B}$ with $\mathbf{B} = \mathbf{K}^{\#}\mathbf{L}^{\#\prime}$
Step 4. Update $\mathbf{C} = \mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})$
Step 5. Finish if convergence is reached; otherwise, back to Step 2.

By substituting $\mathbf{C} = \mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})$ into the loss function (1.18) and using (1.19), the attained value of (1.18) can be expressed as $\|\mathbf{X}^{*} - \mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})(\mathbf{B} \bullet \mathbf{A})'\|^2 = \|\mathbf{X}^{*}\|^2 - \|\mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})\|^2 = \|\mathbf{X}\|^2(1 - GOF_{\mathrm{P}})$, with

$$GOF_{\mathrm{P}} = \frac{\|\mathbf{X}^{*}(\mathbf{B} \bullet \mathbf{A})\|^2}{\|\mathbf{X}\|^2} = \frac{\|\mathbf{C}\|^2}{\|\mathbf{X}\|^2} \tag{1.20}$$

the goodness-of-fit index which takes a value within the range [0, 1] and increases monotonically with the iteration of Steps 2–4. We can use (1.20) for checking the convergence: it can be defined as the change in (1.20) values from the previous round being small enough to be ignored.

1.3.2 Tucker3 Algorithm

For Tucker3, Tucker [33] presented only its approximate solution, and an ALS algorithm for finding the exact solution was later developed by Kroonenberg and de Leeuw [20]. In this algorithm, the Tucker3 model (1.3) or (1.12) is rewritten as

$$\mathbf{X} = \mathbf{A}\mathbf{G}(\mathbf{C} \otimes \mathbf{B})' + \mathbf{E} \tag{1.21}$$

for the block data matrix (1.6), Here, $\mathbf{E}$ contains errors, and $\mathbf{G} = [\mathbf{G}_1, \ldots, \mathbf{G}_R]$ $(P \times RQ)$ contains the elements of the core array (1.4) with $\mathbf{G}_1, \ldots, \mathbf{G}_R$ used in (1.11). Tucker3 is thus formulated as minimizing

$$f_{\mathrm{T3}}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{G}) = \|\mathbf{E}\|^2 = \|\mathbf{X} - \mathbf{A}\mathbf{G}(\mathbf{C} \otimes \mathbf{B})'\|^2. \tag{1.22}$$

Here, we can constrain $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ as

$$\mathbf{A}'\mathbf{A} = \mathbf{I}_P, \mathbf{B}'\mathbf{B} = \mathbf{I}_Q, \mathbf{C}'\mathbf{C} = \mathbf{I}_R \tag{1.23}$$

without loss of generality, since the minimum value of (1.22) is equivalent whether (1.23) is imposed or not, as explained later in Sect. 1.4.

Let us consider minimizing (1.22) over $\mathbf{G}$ for given $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. This problem can be solved by Penrose's [26] regression: the solution is given by

$$\mathbf{G} = \mathbf{A}^{+}\mathbf{X}(\mathbf{C} \otimes \mathbf{B})'^{+} = \mathbf{A}'\mathbf{X}(\mathbf{C} \otimes \mathbf{B}), \tag{1.24}$$

where (1.23) has been used. By substituting (1.24) into (1.22) and using (1.23), we have

$$f_{\mathrm{T3}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{X} - \mathbf{A}\mathbf{A}'\mathbf{X}(\mathbf{C} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{B})'\|^2 = \|\mathbf{X}\|^2 - g(\mathbf{A}, \mathbf{B}, \mathbf{C}) \quad (1.25)$$

with

$$g(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \mathrm{tr}\mathbf{A}'\mathbf{X}(\mathbf{C} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{B})'\mathbf{X}'\mathbf{A} = \mathrm{tr}\mathbf{A}'\mathbf{X}(\mathbf{C}\mathbf{C}' \otimes \mathbf{B}\mathbf{B}')\mathbf{X}'\mathbf{A}. \quad (1.26)$$

It shows that the minimization of (1.25) can be attained by alternately iterating the steps, in each of which (1.26) is maximized over **A**, **B**, or **C** subject to (1.23) with the other two matrices kept fixed.

In the step for **A**, (1.26) is maximized over **A** subject to $\mathbf{A}'\mathbf{A} = \mathbf{I}_P$ with **B** and **C** fixed. It is attained for

$$\mathbf{A} = \mathrm{EV}_P[\mathbf{X}(\mathbf{B}\mathbf{B}' \otimes \mathbf{C}\mathbf{C}')\mathbf{X}'], \quad (1.27)$$

where $\mathrm{EV}_P[\mathbf{M}]$ is the function providing the matrix whose P columns are the eigenvectors corresponding to the largest P eigenvalues of **M**.

In the step for **B**, we arrange the data array (1.2) as (1.16) and the core array (1.4) as $\mathbf{G}^{\#} = [\mathbf{G}_1', \ldots, \mathbf{G}_R']$ $(Q \times RP)$. Then, the model (1.21) can be rewritten as

$$\mathbf{X}^{\#} = \mathbf{B}\mathbf{G}^{\#}(\mathbf{C} \otimes \mathbf{A})' + \mathbf{E}^{\#} \quad (1.28)$$

with $\mathbf{E}^{\#}$ containing errors. It shows that the loss function (1.22) is also rewritten as $f_{\#} = \|\mathbf{X}^{\#} - \mathbf{B}\mathbf{G}^{\#}(\mathbf{C} \otimes \mathbf{A})'\|^2$. Here, we can find that (1.22) becomes equivalent to $f_{\#}$, if **X**, **G**, **A**, and **B** in (1.22) are replaced by $\mathbf{X}^{\#}$, $\mathbf{G}^{\#}$, **B**, and **A**. This replacement can also be made for (1.26) and (1.27) to allow us to find that the optimal **B** to be obtained in this step is

$$\mathbf{B} = \mathrm{EV}_Q[\mathbf{X}^{\#}(\mathbf{A}\mathbf{A}' \otimes \mathbf{C}\mathbf{C}')\mathbf{X}^{\#\prime}]. \quad (1.29)$$

In a parallel manner, the solution in the step for **C** is obtained as follows: the Tucker3 model (1.21) can be rewritten as

$$\mathbf{X}^{*} = \mathbf{C}\mathbf{G}^{*}(\mathbf{B} \otimes \mathbf{A})' + \mathbf{E}^{*}, \quad (1.30)$$

using (1.17) and $\mathbf{G}^{*} = [\mathrm{vec}(\mathbf{G}_1)', \ldots, \mathrm{vec}(\mathbf{G}_R)']$ $(R \times QP)$, which implies that the optimal **C** in this step is given by

$$\mathbf{C} = \mathrm{EV}_R[\mathbf{X}^{*}(\mathbf{B}\mathbf{B}' \otimes \mathbf{A}\mathbf{A}')\mathbf{X}^{*\prime}]. \quad (1.31)$$

We can thus list the algorithm for Tucker3 as follows:

Step 1. Initialize **B** and **C**.
Step 2. Update **A** with (1.27)

Step 3. Update $\mathbf{B}$ with (1.29)
Step 4. Update $\mathbf{C}$ with (1.31)
Step 5. Finish and obtain (1.24) if convergence is reached; otherwise back to Step 2.

The attained value of the loss function (1.25) can be expressed as $f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{X}\|^2(1 - GOF_{\mathrm{T3}})$ with

$$GOF_{\mathrm{T3}} = \frac{g(\mathbf{A}, \mathbf{B}, \mathbf{C})}{\|\mathbf{X}\|^2} = \frac{\|\mathbf{G}\|^2}{\|\mathbf{X}\|^2}. \tag{1.32}$$

Here, we have used (1.24) in (1.26) to have $g(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{G}\|^2$. The value of (1.32), which is within the range [0, 1], expresses the goodness-of-fit for the Tucker3 solution and is convenient for checking the convergence.

Other versions of the Tucker3 algorithm have been developed: ten Berge et al. [30] proposed the algorithm without the constraints in (1.23); Kiers et al. [16] proposed the algorithm in which the inter-column covariance matrix for $\mathbf{X}$ is used rather than $\mathbf{X}$.

1.3.3 Tucker2 Algorithm

The Tucker2 model (1.10) can be rewritten in the following two forms:

$$\mathbf{X} = \mathbf{AH}(\mathbf{I}_K \otimes \mathbf{B})' + \mathbf{E} \tag{1.33}$$
$$\mathbf{X}^{\#} = \mathbf{BH}^{\#}(\mathbf{I}_K \otimes \mathbf{A})' + \mathbf{E}^{\#} \tag{1.34}$$

with $\mathbf{H} = [\mathbf{H}_1, \ldots, \mathbf{H}_K]$ $(P \times KQ)$ and $\mathbf{H}^{\#} = [\mathbf{H}_1', \ldots, \mathbf{H}_K']$ $(P \times KQ)$. Tucker2 is thus formulated as minimzing $\|\mathbf{X} - \mathbf{AH}(\mathbf{I}_K \otimes \mathbf{B})'\|^2 = \|\mathbf{X}^{\#} - \mathbf{BH}^{\#}(\mathbf{I}_K \otimes \mathbf{A})'\|^2$ over $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{H}$ subject to $\mathbf{A}'\mathbf{A} = \mathbf{I}_P$ and $\mathbf{B}'\mathbf{B} = \mathbf{I}_Q$. This problem is the one for Tucker3 with $\mathbf{G}$, $\mathbf{G}^{\#}$, and $\mathbf{C}$ replaced by $\mathbf{H}$, $\mathbf{H}^{\#}$, and $\mathbf{I}_K$, respectively. Applying the replacements to the Tucker3 algorithm can thus lead to the Tucker2 one.

1.4 Rotation of Components

The Tucker3 model has rotational freedom. It is described in Sect. 1.4.1 and the rotation techniques are described in Sects. 1.4.2 and 1.4.3 in which the freedom is exploited for transforming the Tucker3 solution into an interpretable one. Though the Tucker2 model also has the rotational freedom, we do not treat it in this section, as the freedom and the rotation for Tucker2 can be described by those for Tucker3 with the replacements of $\mathbf{G}$, $\mathbf{G}^{\#}$, and $\mathbf{C}$ by $\mathbf{H}$, $\mathbf{H}^{\#}$, and $\mathbf{I}_K$ which were written finally in the last section. The rotational freedom is not possessed by the Parafac model.

1.4.1 Rotational Freedom

The indeterminacy of Tucker3 solutions is shown by that its model (1.21) can be rewritten as

$$\mathbf{X} = \mathbf{AG}(\mathbf{C} \otimes \mathbf{B})' + \mathbf{E} = (\mathbf{AS})\{\mathbf{S}^{-1}\mathbf{G}(\mathbf{U}^{-1} \otimes \mathbf{T}^{-1})'\}\{(\mathbf{CU}) \otimes (\mathbf{BT})\}' + \mathbf{E} \quad (1.35)$$

with $\mathbf{S}$ ($P \times P$), $\mathbf{T}$ ($Q \times Q$), and $\mathbf{U}$ ($R \times R$) arbitrary nonsingular matrices. That is, if $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{G}$ are the optimal solution, then their transformed $\tilde{\mathbf{A}} = \mathbf{AS}$, $\tilde{\mathbf{B}} = \mathbf{BT}$, $\tilde{\mathbf{C}} = \mathbf{CU}$, and $\tilde{\mathbf{G}} = \mathbf{S}^{-1}\mathbf{G}(\mathbf{U}^{-1} \otimes \mathbf{T}^{-1})'$ are also the optimal. This freedom of nonsingular transformation is restricted to that of orthogonal rotation by considering constraint (1.23): $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ can be substituted into $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ in (1.23) only for

$$\mathbf{S}'\mathbf{S} = \mathbf{I}_P, \mathbf{T}'\mathbf{T} = \mathbf{I}_Q, \mathbf{U}'\mathbf{U} = \mathbf{I}_R. \quad (1.36)$$

Then, $\tilde{\mathbf{G}} = \mathbf{S}^{-1}\mathbf{G}(\mathbf{U}^{-1} \otimes \mathbf{T}^{-1})'$ is simplified as in

$$\tilde{\mathbf{A}} = \mathbf{AS}, \quad \tilde{\mathbf{B}} = \mathbf{BT}, \quad \tilde{\mathbf{C}} = \mathbf{CU}, \quad \tilde{\mathbf{G}} = \mathbf{S}'\mathbf{G}(\mathbf{U} \otimes \mathbf{T}). \quad (1.37)$$

However, rotational indeterminacy still remains: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{G}$ can be transformed into $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, and $\tilde{\mathbf{G}}$ as in (1.37) with the rotation matrices $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{U}$ satisfying (1.36).

This indeterminacy can rather be exploited so as to produce interpretable $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, $\tilde{\mathbf{C}}$, and $\tilde{\mathbf{G}}$. The procedures for obtaining such solutions are called orthogonal rotation, which is generally formulated as

$$\text{Optimize a function of (1.37) over } \mathbf{S}, \mathbf{T}, \mathbf{U} \text{ subject to (1.36)}, \quad (1.38)$$

where the function stands for to what extent the argument matrices are interpretable. This interpretability can be defined by that a number of their elements are close to zero, as we may only focus on the elements of large absolute values. Such matrices are said to have simple structure in psychometrics (Thurstone [31]; Trendafilov and Adachi [32]).

A number of orthogonal rotation methods have been proposed, in which the function in (1.38) are defined in different manners (Kroonenberg [19]; Smilde et. al. [25]). Among them, the two methods developed in psychometrics are introduced in Sects. 1.4.2 and 1.4.3. Beside the above orthogonal rotation, a family of the methods called oblique rotation exists, in which (1.36) is relaxed as $\mathbf{S}$, $\mathbf{T}$, $\mathbf{U}$ yielding the component matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ with their columns of unit-length. But, we introduce only the two orthogonal rotation methods which are presented by Kiers [12, 13].

1.4.2 Joint Orthomax Rotation

Kiers [13] has proposed a joint orthomax rotation method. This method is formulated using the orthomax criterion (Crawford and Ferguson [7]) defined for an $L \times M$ matrix $\mathbf{V} = (v_{ij})$ as

$$h_\tau(\mathbf{V}) = \sum_{l=1}^{L}\sum_{m=1}^{M} v_{lm}^4 - \frac{\tau}{L}\sum_{m=1}^{M}\left(\sum_{l=1}^{L} v_{lm}^2\right)^2. \tag{1.39}$$

In particular, (1.39) is called a varimax criterion for $\tau = 1$ (Kaiser [11]). Then, (1.39) stands for the sum of the column variances of squared elements v_{lm}^2. Its maximization provides the matrix $\mathbf{V}$ which has simple structure in that v_{lm}^2 are variant and a number of elements are close to zero.

In the joint orthomax rotation, the function for (1.38) is defined as a weighted sum of (1.39):

$$f_{\mathrm{JO}}(\mathbf{S}, \mathbf{T}, \mathbf{U}) = w_1 h_\tau(\tilde{\mathbf{G}}') + w_2 h_\tau(\tilde{\mathbf{G}}^{\#\prime}) + w_3 h_\tau(\tilde{\mathbf{G}}^{*\prime}) + w_4 h_\tau(\tilde{\mathbf{A}}) + w_5 h_\tau(\tilde{\mathbf{B}}) + w_6 h_\tau(\tilde{\mathbf{C}}). \tag{1.40}$$

Here, $w_1, \ldots, w_6$ are nonnegative weights to be prespecified, and

$$\tilde{\mathbf{G}}^{\#} = \mathbf{T}'\mathbf{G}^{\#}(\mathbf{U} \otimes \mathbf{S}) \text{ and } \tilde{\mathbf{G}}^{*} = \mathbf{U}'\mathbf{G}^{*}(\mathbf{T} \otimes \mathbf{S}) \tag{1.41}$$

are the rotated versions of the matrices $\mathbf{G}^{\#}$ and $\mathbf{G}^{*}$ in the expressions (1.28) and (1.30) for the Tucker3 model. That is, the rotation method is formulated as maximizing (1.40) over $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{U}$ subject to (1.36), in order to jointly give simple structure to component matrices $(\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}})$ and three matrix-expressions of the core array $(\tilde{\mathbf{G}}, \tilde{\mathbf{G}}^{\#}, \tilde{\mathbf{G}}^{*})$. As standard weights, Kiers [12] has suggested $w_1 = v_1 w/(QR)$, $w_2 = v_2 w/(PR)$, $w_3 = v_3 w/(PQ)$, $w_4 = I/P$, $w_5 = J/Q$, and $w_6 = K/R$, with $v_1 = 1/(QR)$, $v_2 = 1/(PR)$, $v_3 = 1/(PQ)$, and $w = PQR/\{(v_1 + v_2 + v_3)\|\mathbf{G}\|^4\}$.

The solution of $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{U}$ is obtained by alternately iterating the steps, in each of which (1.40) is maximized over each of the three matrices with the other two ones kept fixed. In the step for $\mathbf{S}$, the part of (1.41) relevant to $\mathbf{S}$, i.e., $g_{\mathrm{JO}}(\mathbf{S}) = w_1 h_\tau(\tilde{\mathbf{G}}') + w_2 h_\tau(\tilde{\mathbf{G}}^{\#\prime}) + w_3 h_\tau(\tilde{\mathbf{G}}^{*\prime}) + w_4 h_\tau(\tilde{\mathbf{A}})$, is maximized over $\mathbf{S}$ with $\mathbf{T}$ and $\mathbf{U}$ fixed, where $g_{\mathrm{JO}}(\mathbf{S})$ can be simplified as

$$g_{\mathrm{JO}}(\mathbf{S}) = w^* h_{\tau^*}(\tilde{\mathbf{G}}') + w_4 h_\tau(\tilde{\mathbf{A}}) = w^* h_{\tau^*}(\mathbf{G}_{\mathrm{UT}}'\mathbf{S}) + w_4 h_\tau(\mathbf{AS}), \tag{1.42}$$

with $w^* = w_1 + w_2 + w_3$, $\tau^* = \tau w_1/w^*$, and $\mathbf{G}_{\mathrm{UT}} = \mathbf{G}(\mathbf{U} \otimes \mathbf{T})$ (Kiers [12]). For the maximization of (1.42), the fact is used that any orthonormal $\mathbf{S}$ is reparameterized as

$$\mathbf{S} = \prod_{y=1}^{Y}\prod_{z=y+1}^{Z} \mathbf{S}_{yz} \tag{1.43}$$

Here, $Y < Z \leq P$, and $\mathbf{S}_{yz}$ is the $P \times P$ identity matrix with its (y, y), (z, z), (y, z), and (z, y) elements replaced by $\cos\theta$, $\cos\theta$, $-\sin\theta$, and $\sin\theta$, respectively: $\mathbf{S}_{yz}$ is determined by a single angle θ. Equation (1.43) shows that (1.42) increases monotonically by repeating

$$\max_{\theta} g_{\mathrm{JO}}(\theta) = w^{*} h_{\tau *}(\mathbf{G}'_{yz}\mathbf{S}_{yz}) + w_4 h_{\tau}(\mathbf{A}_{yz}\mathbf{S}_{yz}) \tag{1.44}$$

with increasing y and z as $[y, z] = [1, 2], [1, 3], \ldots, [Y, Z-1], [Y, Z], \ldots, [P-1, P]$. Here, $\mathbf{G}_{yz} = \tilde{\mathbf{G}}'_{12} \prod_{[y',z'] \in \Phi} \mathbf{S}_{y'z'}$, and $\mathbf{A}_{yz} = \tilde{\mathbf{A}}_{12} \prod_{[y',z'] \in \Phi} \mathbf{S}_{y'z'}$ with Φ denoting a set of $[y, z]'$s preceding the current $[y, z]$, $\tilde{\mathbf{G}}_{12}$ being the matrix $\tilde{\mathbf{G}}$ before the start of repeating (1.44), and $\tilde{\mathbf{A}}_{12}$ the matrix $\tilde{\mathbf{A}}$ before the start. By iterating the repetition of (1.44) until convergence is reached, (1.42) can be maximized. The procedure for (1.44) is described in Kiers [12]. Shiba [24] has also detailed the maximization of $h_{\tau}(\mathbf{A}_{yz}\mathbf{S}_{yz})$ over θ. In parallel manners, the algorithms in the steps for $\mathbf{T}$ and $\mathbf{U}$ can be formed.

1.4.3 Three-Way Simplimax Rotation

This rotation method has also been developed by Kiers [13]. In the method, only simple structure of core array is considered, and $\tilde{\mathbf{G}}$ is obtained so as to be matched with a $P \times RQ$ target matrix $\mathbf{G}_{\mathrm{T}}$ which includes a specified number of zero elements. The method has both orthogonal and oblique versions. Here, we only describe the former, i.e., the orthogonal three-way simplimax rotation. It is formulated as

$$\min_{\mathbf{S},\mathbf{T},\mathbf{U},\mathbf{G}_{\mathrm{T}}} \|\tilde{\mathbf{G}} - \mathbf{G}_{\mathrm{T}}\|^2 \text{ subject to (1.36) and } N_0(\mathbf{G}_{\mathrm{T}}) = \kappa \tag{1.45}$$

where $N_0(\mathbf{G}_{\mathrm{T}})$ denotes the number of zero elements in $\mathbf{G}_{\mathrm{T}}$ and κ is a specified integer. An interesting feature of the problem (1.45) is that the target $\mathbf{G}_{\mathrm{T}}$ is also to be estimated: only its number of zero elements $N_0(\mathbf{G}_{\mathrm{T}})$ is known to be κ, thus the locations of those elements are to be estimated with the values of nonzero elements.

The problem (1.45) can be attained by alternately iterating the steps, in each of which $\|\tilde{\mathbf{G}} - \mathbf{G}_{\mathrm{T}}\|^2$ is minimized over $\mathbf{S}$, $\mathbf{T}$, $\mathbf{U}$, or $\mathbf{G}_{\mathrm{T}}$ with the remaining three matrices fixed. The solution in the step for $\mathbf{G}_{\mathrm{T}}$, i.e., the optimal $\mathbf{G}_{\mathrm{T}} = (g_{pn}^{[\mathrm{T}]})$ that minimizes $\|\tilde{\mathbf{G}} - \mathbf{G}_{\mathrm{T}}\|^2$ with $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{U}$ fixed, is given by $g_{pn}^{[\mathrm{T}]} = 0$ if $|g_{pn}^{[\mathrm{T}]}| \leq |g_{\kappa}|$ and by $g_{pn}^{[\mathrm{T}]} = \tilde{g}_{pn}$ otherwise. Here, $\tilde{g}_{pn}$ is the (p, n) element of $\tilde{\mathbf{G}}$ and $|g_{\kappa}|$ denotes the κth smallest one among the absolute values of the elements in $\tilde{\mathbf{G}} = (\tilde{g}_{pn})$. The solution in the step for $\mathbf{S}$, i.e., the $\mathbf{S}$ that minimizes $\|\tilde{\mathbf{G}} - \mathbf{G}_{\mathrm{T}}\|^2 = \|\mathbf{S}'\mathbf{G}(\mathbf{U} \otimes \mathbf{T}) - \mathbf{G}_{\mathrm{T}}\|^2 = \|\mathbf{G}(\mathbf{U} \otimes \mathbf{T})\|^2 + \|\mathbf{G}_{\mathrm{T}}\|^2 - 2\mathrm{tr}\mathbf{S}'(\mathbf{U} \otimes \mathbf{T})\mathbf{G}'_{\mathrm{T}}$ for given $\mathbf{G}_{\mathrm{T}}$, $\mathbf{T}$, and $\mathbf{U}$, is obtained by $\mathbf{S} = \Gamma\Xi'$, where Γ and Ξ are obtained through the SVD $(\mathbf{U} \otimes \mathbf{T})\mathbf{G}'_{\mathrm{T}} = \Gamma\Sigma\Xi'$ with Σ the diagonal matrix including singular values. In parallel manners, the steps for $\mathbf{T}$ and $\mathbf{U}$ can be formed.

1.5 Applications to Stimulus-Response Data

In this section, we illustrate 3WPCA with stimulus-response data observed in psychology. In general, a set of those data takes the form of an I-stimuli $\times$ J-responses $\times$ K-persons array $\{x_{ijk}\}$ with x_{ijk} indicating the strength of response j evoked by person k against stimulus i (Adachi [3]; Kiers and Van Mechelen [18]). For example, stimuli are various auditory signals, responses are bodily reactions and impression ratings. We first introduce the graphical representation of 3WPCA models for such a data set, which is followed by the illustration of 3WPCA with a real data example.

1.5.1 Network Representations of Three-Way PCA

The Tucker2 model for I-stimuli $\times$ J-responses $\times$ K-persons data can be represented as a network diagram (Adachi [3]) with its constrained versions the diagrams for Tucker3 and Parafac.

Recall the Tucker2 model (1.10), i.e., $\mathbf{X}_k = \mathbf{A}\mathbf{H}_k\mathbf{B}' + \mathbf{E}_k$. In this section, $\mathbf{X}_k = (x_{ijk})$ is an I-stimuli $\times$ J-responses data matrix containing the response strengths for person k. The network representation for the model is illustrated in Fig. 1.2 with $I = 6$, $J = 5$, $P = 3$, and $Q = 2$. The equivalence of the diagram to the Tucker2 model (1.10) is explained in the next paragraph.

As indicated by the links in the figure, the (i, p) element of $\mathbf{A} = (a_{ip})$ represents the association of stimulus i to *stimulus component* p, the (j, q) element of $\mathbf{B} = (b_{jq})$ represents that of *response component* q to response j, and the (p, q)th element of $\mathbf{H}_k = (h_{pqk})$ expresses the association of stimulus component p with response component q. Here, let i-p-q-j express the route starting at stimulus i to reach response j by way of components p and q, and let $\eta^{[k]}_{i\text{-}p\text{-}q\text{-}j}$ denote the effect that

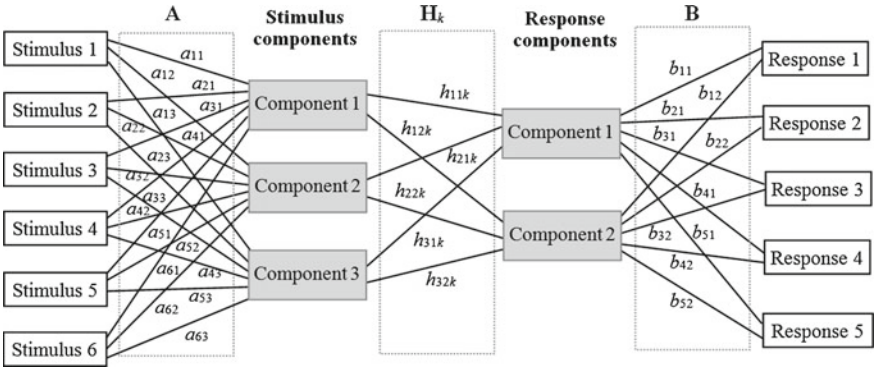


Fig. 1.2 Network representation of the Tucker2 model for person k with $I = 6$, $J = 5$, $P = 3$, and $Q = 2$

stimulus i causes through the route i-p-q-j on the response j of person k. The equivalence of the network and the model (1.10) follows from assuming that (1) the effect $\eta^{[k]}_{i\text{-}p\text{-}q\text{-}j}$ is given by the product of the associations indicated by the links of i to p, p to q, and q to j on the route i-p-q-j and that (2) the total effect of stimulus i on response j for person k, which approximates x_{ijk}, is expressed as the sum of the effects corresponding to all the routes connecting stimulus i to response j, that is, the sum of $\eta^{[k]}_{i\text{-}p\text{-}q\text{-}j}$ over $p = 1, \ldots, P$ and $q = 1, \ldots, Q$. Assumption (1) leads to $\eta^{[k]}_{i\text{-}p\text{-}q\text{-}j} = a_{ip}h_{pqk}b_{jq}$ and (2) leads to $x_{ijk} \cong \sum_p \sum_q \eta^{[k]}_{i\text{-}p\text{-}q\text{-}j} = \sum_p \sum_q a_{ip}h_{pqk}b_{jq}$, which is written as (1.10) in a matrix form.

In (1.10), $\mathbf{H}_k = (h_{pqk})$ has subscript k, while it is not attached to $\mathbf{A} = (a_{ip})$ and $\mathbf{B} = (b_{jq})$. Thus, the links of stimuli to their components and those of responses in Fig. 1.2 are invariant across persons, while the strengths (h_{pqk}) of the inter-component links vary across persons $k = 1, \ldots, K$. It implies that K inter-component links with different h_{pqk} must be considered. However, Parafac and Tucker3 which are the version of Tucker2 with $\mathbf{H}_k$ constrained lead to inter-component links easily interpreted, which is illustrated in Sects. 1.5.3 and 1.5.4 using the data set described next.

1.5.2 Color-Adjective Data

The stimulus-response data $\{x^{(R)}_{ijk}\}$ available at the web page http://bm.hus.osaka-u.ac.jp/data/ is used as an example. In this data set, stimuli and responses are eleven color names and nine adjectives, respectively, (see Table 1.2) with the number of persons thirty: $x^{(R)}_{ijk}$ is the rating value indicating how strongly color i gives the impression expressed by adjective j, with the rating made by person k. As found in the web page, $x^{(R)}_{ijk}$ being four stands for a neutral impression, and thus we subtracted four from $x^{(R)}_{ijk}$ to have the data set $\{x_{ijk}; x_{ijk} = x^{(R)}_{ijk} - 4\}$ to be analyzed. Parts of the resulting $\mathbf{X}_k = (x_{ijk})$ for $k = 1, 30$ are presented in Fig. 1.3. There, for example, the (*brown-blight*) element in $\mathbf{X}_1$ is found to be -2, which implies that *brown* impressed person 1 as rather "*not blight*" or "*dark*". We perform Parafac and Tucker3 for those $\mathbf{X}_k$, $k = 1, \ldots, 30$.

$\mathbf{X}_1$

	heavy	*bright*	...	*strong*
brown	1	-2	...	0
y-green	-1	0	...	0
pink	-3	1	...	-3
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
orange	-1	2	...	1
red	1	2	...	1

...

$\mathbf{X}_{30}$

	heavy	*bright*	...	*strong*
brown	2	-2	...	0
y-green	-1	2	...	0
pink	0	2	...	-1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
orange	0	2	...	1
red	0	3	...	1

Fig. 1.3 Stimuli × responses data matrices $\mathbf{X}_k$ for persons $k = 1, \ldots, 30$

1.5.3 Parafac Solution

Recall the Parafac model (1.13), i.e., $\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{B}' + \mathbf{E}_k$: the matrix $\mathbf{H}_k$ are constrained to equal a $P \times P$ diagonal matrix $\mathbf{D}_k = \text{diag}(c_{k1}, \ldots, c_{kP})$ with $P = Q$. We fitted the Parafac model to the color-adjective data with $P = Q = 3$. This number was chosen, as it is facilitated to compare the Parafac solution with the Tucker3 one described in the next section, where $P = Q = 3$ is chosen according to the results of preliminary analysis. As described in Sect. 1.3, Parafac without constraint (1.19) tends to provide a degenerated solution. Indeed, such a solution was also obtained for the color-adjective data. We thus reported only the solution subject to constraint (1.19). The resulting GOF value of (1.20) was 0.50. Table 1.1 presents the resulting **A** and **B**, whose columns, i.e., components, are indicated by the labels, *pale*, …, *impact*, standing for the interpretations of the components. Why such interpretation can be made would be explained later. On the other hand, the solution for c_{k1}, c_{k2}, and c_{k3} ($k = 1, \ldots, 30$) is shown in Fig. 1.4, where $c_{1p}, \ldots, c_{30p}$ are connected by lines for each p.

Table 1.1 Parafac solution of **A** and **B**

A (colors × components)				**B** (adjectives × components)			
Color	*Pale*	*Blue*	*Primary*	Adjective	*Good*	*Inactive*	*Impact*
Brown	−0.41	−0.02	0.01	*Heavy*	−0.44	0.17	0.24
Yellow green	0.32	0.03	0.00	*Bright*	0.45	−0.31	0.21
Pink	0.38	−0.16	0.04	*Beautiful*	0.44	0.10	0.53
Yellow	0.25	−0.28	0.12	*Muddy*	−0.44	−0.18	−0.20
Sky blue	0.58	0.37	−0.01	*Noisy*	−0.13	−0.59	−0.04
Navy	−0.28	−0.31	0.21	*Hot*	−0.10	−0.60	0.09
Purple	−0.24	0.32	0.31	*Distant*	−0.12	0.24	−0.04
Blue	0.18	0.44	0.45	*Deep*	−0.37	0.16	0.53
Green	0.06	0.19	0.38	*Strong*	−0.21	−0.19	0.54
Orange	0.14	−0.30	0.18				
Red	−0.04	−0.49	0.68				

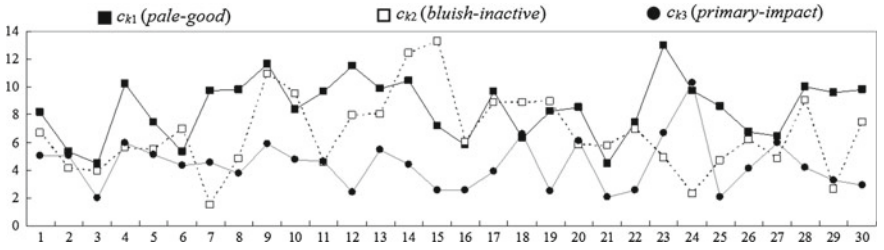


Fig. 1.4 Parafac solution for c_{kp} as a function of person $k = 1, \ldots, 30$

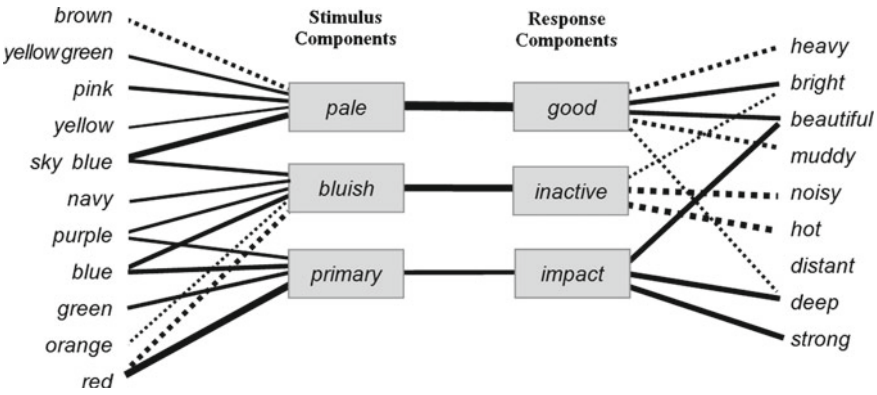


Fig. 1.5 Network representation of the Parafac solution

Since of $P = Q$ and $\mathbf{H}_k$ being a diagonal matrix $\mathbf{D}_k$, the network diagram in Fig. 1.2 is simplified in Parafac so that the number of stimulus components are equal to the number of response ones and they are simply connected by parallel links. The Parafac solution in Table 1.1 can be represented as in Fig. 1.5, where the links are depicted with the following rules:

(R_1) The links are omitted that correspond to the parameters with their absolute values less than 0.3.
(R_2) The widths of the links for colors and adjectives are proportional to the absolute values of the corresponding parameter values.
(R_3) The real and dashed lines indicate positive and negative values, respectively.
(R_4) Each inter-component link is proportional to the average $K^{-1}\sum_{k=1}^{K} c_{kp}$ of $K = 30$ persons.

First, let us note the links of colors to stimulus components in Fig. 1.4. Pale colors are found to be positively associated with the first component, which allows us to interpret it as the "sensor" that senses *pale* colors. In a parallel manner, the second component can be interpreted as the sensor for *bluish* colors. The third one can be named as the component sensing *primary* colors, as it is associated strongly with *red* and positively with *blue* that is also primary as red.

Next, let us note the response components. They can be interpreted in line with Osgood et al. [23] psychological theory on adjectives: they have argued that every adjective can be classified into one of the three groups G1, G2, and G3, where the adjectives in G1 involve evaluation, i.e., the *goodness* of impression, the ones in G2 expresses *activity*, and the adjectives in G3 stand for *potency*. The first and second components clearly correspond to G1 and G2, respectively. But, the adjectives related to *activity* are negatively related to the second component, thus we label it as *inactive*. Though G3 can be related to the third component, we rather name it *impact*, as the adjective *beautiful*, which seems not to involve potency, also shows strong association.

The inter-component links in Fig. 1.5 show that *pale*, *bluish*, and *primary* colors evoke *good*, *inactive*, and *impact*-inducing impressions, respectively. Among those associations, *pale-good* is the strongest and *primary-impact* is the weakest in average. But, the strengths of the associations are found to differ across persons in Fig. 1.4: for example, in the 15th person, *bluish-inactive*is rather the strongest.

1.5.4 Tucker3 Solution

Recall (1.11), i.e., $\mathbf{H}_k = c_{k1}\mathbf{G}_1 + \cdots + c_{kR}\mathbf{G}_R$ in Tucker3. We set $R = 2$ and $P = Q = 3$ following the results of the preliminary analysis by Adachi [1] and carried out Tucker3 for the color-adjective data. The resulting GOF value of (1.33) was 0.52. The solution was rotated by Kiers' [12] joint orthomax method with $\tau = 1$ and w_l set to the standard weights. The resulting $\mathbf{A}$, $\mathbf{B}$, $\mathbf{G}_1$, and $\mathbf{G}_2$ are shown in Table 1.2, while the c_{k1} and c_{k2} values for $k = 1, \ldots, 30$ are connected by lines in Fig. 1.6. Since of $R = 2$, (1.11) is simplified as $\mathbf{H}_k = c_{k1}\mathbf{G}_1 + c_{k2}\mathbf{G}_2$: the matrix $\mathbf{H}_k$ describing the inter-component links for person k is a weighted sum of $\mathbf{G}_1$ and $\mathbf{G}_2$ which are invariant across k. With taking it in account, the weights c_{k1} for $\mathbf{G}_1$ should be noted in Fig. 1.6: c_{k1} are positive for all k. It implies that $\mathbf{G}_1$ stands for the inter-component links common over persons. On the other hand, the persons are split by the signs of the weights c_{k2} for $\mathbf{G}_2$, which implies that $\mathbf{G}_2$ stands for

Table 1.2 Tucker3 solution of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{G}_r$

$\mathbf{A}$ (colors × components)				$\mathbf{B}$ (adjectives × components)			
	Pale	*Blue*	*Red*		*Good*	*Inactive*	*Impact*
Brown	−0.39	−0.11	0.00	*Heavy*	−0.47	0.11	0.22
Yellow green	0.30	0.10	−0.06	*Bright*	0.50	−0.13	0.28
Pink	0.40	−0.03	0.07	*Beautiful*	0.41	0.33	0.44
Yellow	0.30	−0.13	0.22	*Muddy*	−0.38	−0.30	−0.13
Sky blue	0.48	0.44	−0.23	*Noisy*	0.01	−0.61	0.11
Navy	−0.36	0.33	0.09	*Hot*	0.06	−0.57	0.23
Purple	−0.32	0.33	0.10	*Distant*	−0.15	0.17	−0.11
Blue	0.05	0.61	0.16	*Deep*	−0.40	0.19	0.49
Green	−0.01	0.39	0.29	*Strong*	−0.18	−0.07	0.59
Orange	0.21	−0.13	0.28				
Red	0.04	−0.12	0.83				
$\mathbf{G}_1$ (color-components × adj-components)				$\mathbf{G}_2$ (color-components × adj-components)			
	Good	*Inactive*	*Impact*		*Good*	*Inactive*	*Impact*
Pale	47.94	0.16	−1.61	*Pale*	−2.25	−3.91	5.62
Blue	0.04	34.22	2.90	*Blue*	12.47	6.82	−6.20
Red	1.56	−11.26	25.03	*Red*	10.39	−3.88	0.40

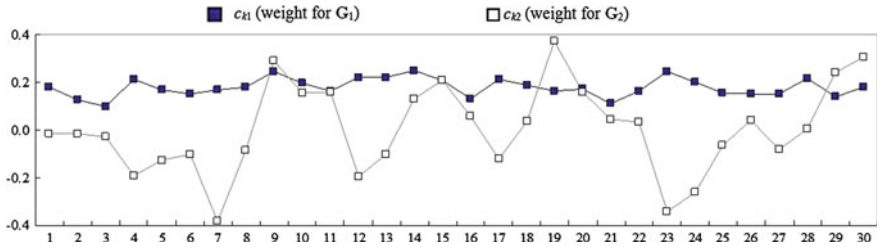


Fig. 1.6 Tucker3 solution for ckp as a function of person $k = 1, \ldots, 30$

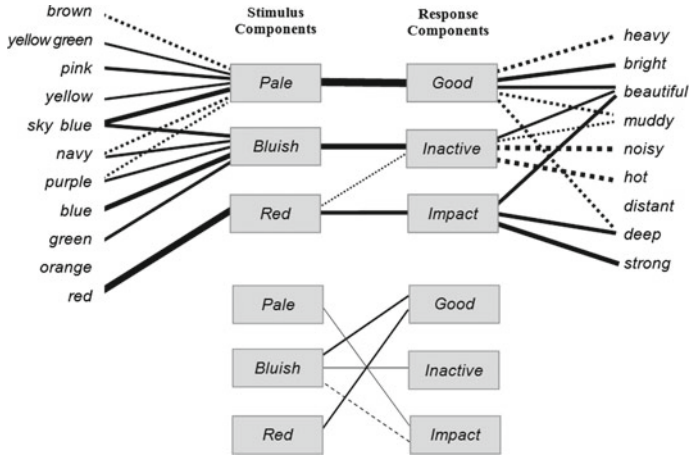


Fig. 1.7 Network representation of the Tucker3 solution

individual differences across persons. Those arguments allow us to represent the Tucker3 solution as in Fig. 1.7. There, the upper diagram in which inter-component links correspond to $\mathbf{G}_1$ stands for the network common over persons, while the lower diagram, the links corresponding to $\mathbf{G}_2$, expresses the individual differences in inter-component links. The rules by which Fig. 1.7 was depicted are the same as (R1), (R2), (R3) in Sect. 1.5.3, except the following point: the links have been omitted that correspond to the elements of $\mathbf{G}_1$ and $\mathbf{G}_2$ less than five in Fig. 1.7.

The stimulus and response components in Fig. 1.7 can be given the same labels as those for the Parafac solution, except for that the third stimulus component in Fig. 1.7 is interpreted simply as the sensor for *red*, as only that color is strongly associated with the third component. The inter-component links corresponding to $\mathbf{G}_1$ are found to be similar to the Parafac one, in that parallel links are dominant, except for that the unparallel (negative) link of *red* to *inactive* is also found.

As described above, the matrix $\mathbf{G}_2$ in $\mathbf{H}_k = c_{k1}\mathbf{G}_1 + c_{k2}\mathbf{G}_2$ stands for the individual differences in inter-component links. For example, in Fig. 1.6, $|c_{72}|$ for person 7 is found to be large with $c_{71} = 0.17$ and $c_{72} = -0.38$, the links for that person

are expressed as $\mathbf{H}_k = 0.17\mathbf{G}_1 - 0.38\mathbf{G}_2$. On the other hand, c_{k2} for $k = 1, 2, 3$ are close to zero, thus $\mathbf{H}_k$ is almost proportional to $\mathbf{G}_1$.

1.6 Conclusions

In this chapter, we described the popular 3WPCA procedures, Tucker2, Tucker3, and Parafac, which are designed for analyzing a three-way data array $\dddot{\mathbf{X}} = \{x_{ijk}; i = 1, \ldots, I; j = 1, \ldots, J; k = 1, \ldots, K\}$. This array can be rearranged in an $I \times JK$ block matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_K]$ whose kth block $\mathbf{X}_k$ includes x_{ijk} as its (i, j) element. According as how to express three-way data, the models for 3WPCA can be written in different forms. In Table 1.3, the four forms used in this chapter are listed. There, $\circ$, $\otimes$, and $\bullet$ denote the tensor, Kronecker, and Khatri-Rao products, respectively, with $\mathbf{h}_{pq} = [h_{pq1}, \ldots, h_{pqK}]'$ $(K \times 1)$.

The four forms in Table 1.3 were used for different purposes in this chapter. The scalar and tensor forms were used for introducing Tucker3 and Parafac in Sect. 1.1. On the other hand, the matrix expressions with $\mathbf{X}_k$ were mainly used for describing the hierarchical relationships among 3WPCA and the ordinary PCA in Sect. 1.2. Those expressions were also used for the network representations of 3WPCA models in Sect. 1.5. The other matrix expression is the one with a block matrix $\mathbf{X}$, which is useful for describing the algorithms for 3WPCA and rotation methods, as found in Sects. 1.3 and 1.4.

In Sect. 1.5, we illustrated the network representations of 3WPCA models for "stimuli × responses × persons" data. Here, the data description can be replaced by "inputs × outputs × black-boxes" to extend the areas in which 3WPCA is useful. In various areas including medicine and engineering, the data are observed that describe the outputs of black-boxes against input signals. 3WPCA allows us to visually capture the components and their mutual links in the black-boxes with the network representations as in Figs. 1.2, 1.5, and 1.7. The data for which 3WPCA is usefully performed are not restricted to such inputs × outputs × black-boxes ones. The applications to other types of data have been illustrated in Kroonenberg [19] and Simlde et al. [25].

Table 1.3 Four expressions of data and the corresponding model parts

	Data	Tucker2	Tucker3	Parafac
1. Scalar	x_{ijk}	$\sum_{p=1}^{P}\sum_{q=1}^{Q} a_{ip}b_{jq}h_{pqk}$	$\sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a_{ip}b_{jq}c_{kr}g_{pqr}$	$\sum_{p=1}^{P} a_{ip}b_{jp}c_{kp}$
2. Tensor	$\dddot{\mathbf{X}}$	$\sum_{p=1}^{P}\sum_{q=1}^{Q} \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{h}_{pq}$	$\sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} (\mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r)g_{pqr}$	$\sum_{p=1}^{P} \mathbf{a}_p \circ \mathbf{b}_p \circ \mathbf{c}_p$
3. Matrix	$\mathbf{X}_k$	$\mathbf{A}\mathbf{H}_k\mathbf{B}'$	$\mathbf{A}\sum_{r=1}^{R} c_{kr}\mathbf{G}_r\mathbf{B}'$	$\mathbf{A}\mathbf{D}_k\mathbf{B}'$
4. Matrix	$\mathbf{X}$	$\mathbf{A}\mathbf{H}(\mathbf{I}_k \otimes \mathbf{B})'$	$\mathbf{A}\mathbf{G}(\mathbf{C} \otimes \mathbf{B})'$	$\mathbf{A}(\mathbf{C} \bullet \mathbf{B})'$

References

1. Adachi, K. (2008). Studies on joint Procrustes analysis of three-way data. Reports for the studies supported by the Grand-in-Aid for Scientific Research (Grant (C)-18500212 from the Japan Society for the Promotion of Science).
2. Adachi, K. (2009). Joint Procrustes analysis for simultaneous nonsingular transformation of component score and loading matrices. *Psychometrika*, *74*, 667–683. doi:10.1007/s11336-009-9131-2.
3. Adachi, K. (2011). Three-way Tucker2 component analysis solutions of stimuli × responses × individuals data with simple structure and the fewest core differences. *Psychometrika*, *76*, 285–305. doi:10.1007/s11336-011-9208-6.
4. Adachi, K. (2013). Generalized joint Procrustes analysis. *Computational Statistics*, *28*, 2449–2464. doi:10.1007/s00180-013-0413-x.
5. Carroll, J. D., & Chang, J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young" decomposition. *Psychometrika*, *35*, 282–319. doi:10.1007/BF02310791.
6. Ceulemans, E., & Kiers, H. A. L. (2006). Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology*, *59*, 133–150. doi:10.1348/000711005X64817.
7. Crawford, C. B., & Ferguson, G. A. (1970). A general rotation criterion and its use in orthogonal rotation. *Psychometrika*, *35*, 321–332. doi:10.1007/BF02310792.
8. Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an “exploratory" multi-mode factor analysis. *UCLA Working Papers in Phonetics*, *16*, 1–84.
9. Harville, D. A. (1997). *Matrix algebra from a statistician's perspective*. New York: Springer.
10. Hitchcock, F. L. (1927). Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematical Physics*, *7*, 39–79.
11. Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, *23*, 187–200. doi:10.1007/BF02289233.
12. Kiers, H. A. L. (1998a). Joint orthomax rotation of the core and component matrices resulting from three-mode principal component analysis. *Journal of Classification*, *15*, 245–263. doi:10.1007/s003579900033.
13. Kiers, H. A. L. (1998b). Three-way SIMPLIMAX for oblique rotation of the three-mode factor analysis core to simple structure. *Computational Statistics & Data Analysis*, *28*, 307–324. doi:10.1016/S0167-9473(98)00041-3.
14. Kiers, H. A. L. (1991). Hierarchical relations among three-way methods. *Psychometrika*, *56*, 449–470. doi:10.1007/BF02294485.
15. Kiers, H. A. L., & Krijnen, W. P. (1991). An efficient algorithm for Parafac of three-way data with large numbers of observation units. *Psychometrika*, *56*, 147–152. doi:10.1007/BF02294592.
16. Kiers, H. A. L., Kroonenberg, P. M., & ten Berge, J. M. F. (1992). An efficient algorithm for Tuckals3 on data with large numbers of observation units. *Psychometrika*, *57*, 415–422. doi:10.1007/BF02295429.
17. Kiers, H. A. L., & ten Berge, J. M. F. (1994). Hierarchical relations between methods for simultaneous component analysis and a technique for rotation to a simple simultaneous structure. *British Journal of Mathematical and Statistical Psychology*, *47*, 109–126. doi:10.1111/j.2044-8317.1994.tb01027.x.
18. Kiers, H. A. L., & Van Mechelen, I. (2001). Three-way component analysis: Principles and illustrative application. *Psychological Methods*, *6*, 84–110. doi:10.1037/1082-989X.6.1.84.
19. Kroonenberg, P. M. (2008). *Applied multiway data analysis*. Hoboken: Wiley.
20. Kroonenberg, P. M., & de Leeuw, J. (1980). Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, *45*, 69–97. doi:10.1007/BF02293599.

21. Mulaik, S. A. (2010). *Foundations of factor analysis* (2nd ed.). Boca Raton: CRC Press.
22. Murakami, T., & Kroonenberg, P. M. (2003). Three-mode models and individual differences in semantic differential data. *Multivariate Behavioral Research*, *38*, 247–283. doi:10.1207/S15327906MBR3802_5.
23. Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). *The measurement of meaning*. Urbana, IL: University of Illinois Press.
24. Shiba, S. (1979). *Factor analysis* (2nd ed.). Tokyo: University of Tokyo Press. (in Japanese).
25. Smilde, A., Bro, R., & Geladi, P. (2004). *Multi-way analysis: Applications in the chemical sciences*. Chichester: Wiley.
26. Penrose, R. (1956). On the best approximate solutions of linear matrix equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, *52*, 17–19.
27. Rao, C. R., & Mitra, S. K. (1971). *Generalized inverse of matrices and its applications*. New York: Wiley.
28. ten Berge, J. M. F. (1993). *Least squares optimization in multivariate analysis*. Leiden: DSWO Press.
29. ten Berge, J. M. F., & Kiers, H. A. L. (1996). Optimality criteria for principal component analysis and generalizations. *British Journal of Mathematical and Statistical Psychology*, *49*, 335–345. doi:10.1111/j.2044-8317.1996.tb01092.x.
30. ten Berge, J. M. F., de Leeuw, J., & Kroonenberg, P. M. (1987). Some additional results on principal components analysis of three-mode data by alternating least squares algorithms. *Psychometrika*, *52*, 183–191. doi:10.1007/BF02294233.
31. Thurstone, L. L. (1935). *The vectors of mind*. Chicago: University of Chicago Press.
32. Trendafilov, N. T., & Adachi, K. (2014). Sparse vs. simple structure loadings. *Psychometrika*. doi:10.1007/s11336-014-9416-y (to appear).
33. Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, *31*, 279–311. doi:10.1007/BF02289464.

Chapter 2
Non-negative Matrix Factorization and Its Variants for Audio Signal Processing

Hirokazu Kameoka

Abstract In this chapter, I briefly introduce a multivariate analysis technique called non-negative matrix factorization (NMF), which has attracted a lot of attention in the field of audio signal processing in recent years. I will mention some basic properties of NMF, effects induced by the non-negative constraints, how to derive an iterative algorithm for NMF, and some attempts that have been made to apply NMF to audio processing problems.

Keywords Non-negative matrix factorization · Majorization-minimization algorithm · Bregman divergence · Bayesian nonparametrics · Audio signal processing

2.1 Introduction

There are many kinds of real-world data given by non-negative values, such as power spectra, pixel values and count data. In a way similar to multivariate analysis techniques such as Principal Component Analysis (PCA) and Independent Component Analysis (ICA), decomposing non-negative data into the sum of the underlying components can be useful in many situations: For example, if we can extract the power spectra of the underlying sources in a mixture signal, they may be useful for noise reduction and source separation. If we can decompose face images into components corresponding to facial features such as the eyes, nose and mouth, they may be useful for face recognition, identification and synthesis. If we can decompose the word histograms of text documents into components associated with latent topics such as politics, sport and economy, they may be useful for document indexing and retrieval. Similarly, if we can extract patterns reflecting users' preferences from purchase logs,

H. Kameoka (✉)
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
e-mail: kameoka@hil.t.u-tokyo.ac.jp; kameoka.hirokazu@lab.ntt.co.jp

H. Kameoka
Nippon Telegraph and Telephone Corporation, 3-1 Morinosato Wakamiya, Atsugi, Kanagawa 243-0198, Japan

T. Sakata (ed.), *Applied Matrix and Tensor Variate Data Analysis*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55387-8_2

they may be useful for making recommendations. A multivariate analysis technique enabling the decomposition of non-negative data into non-negative components is called Non-negative Matrix Factorization (NMF) [1]. In this chapter, I will mention some basic properties of NMF, how to derive an iterative algorithm for NMF, and some attempts that have been made to apply NMF and its variants to audio processing problems.

2.2 What Is NMF?

In the following, we will represent data by vectors. For image data, each pixel value will correspond to a single element of the data vector. For power spectrum data, the power at each frequency point will correspond to a single element of the data vector. Let us assume that we are given a set of N non-negative data vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \in \mathbb{R}^{\geq 0, K}$. We refer to each of them as an observed vector. Here, $\mathbb{R}^{\geq 0, K}$ is used to represent an entire set of K-dimensional non-negative vectors. The aim of NMF is to decompose each of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ into the sum of M non-negative components: The problem is to find the linear combinations of M basis vectors $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M \in \mathbb{R}^{\geq 0, K}$ that best approximate $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$:

$$\boldsymbol{y}_n \simeq \sum_{m=1}^{M} \boldsymbol{h}_m u_{m,n} \quad (n = 1, \ldots, N), \tag{2.1}$$

subject to the non-negativity constraints on both the basis vectors $\boldsymbol{h}_m$ and the coefficients $u_{m,n}$. Here, it is important to note that the observed data are assumed to be quantities that are additive in nature. Although neither a pixel value nor a power spectrum is strictly an additive quantity, we must be aware of the fact that when applying NMF, the additivity of the data of interest will be implicitly assumed to hold, regardless of whether this assumption is true or only approximately true. The non-additivity of power spectra will be discussed in detail in Sect. 2.7. In addition to the additivity assumption as regards the data, the non-negativity constraint is one of the most important features of NMF. As explained later, the non-negativity constraint contributes to inducing sparsity of both the basis vectors and the coefficients.

Now, if we let $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N] = (y_{k,n})_{K \times N}$, $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M] = (h_{k,m})_{K \times M}$ and $\boldsymbol{U} = (u_{m,n})_{M \times N}$, Eq. (2.1) can be rewritten as $\boldsymbol{Y} \simeq \boldsymbol{H}\boldsymbol{U}$. NMF can thus be seen as a problem of factorizing an observed matrix into the product of two non-negative matrices, which gives NMF its name. To understand NMF intuitively, see Fig. 2.1 for an example of NMF applied to the spectrogram of an audio signal, interpreted as a non-negative matrix.

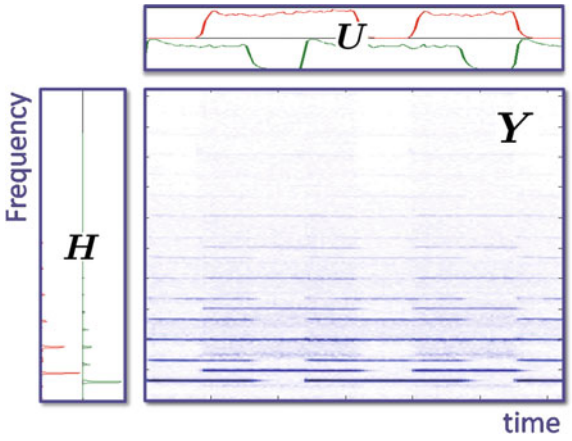


Fig. 2.1 NMF applied to the spectrogram of an audio signal. Each column of $\boldsymbol{H}$ and each row of $\boldsymbol{U}$ can be interpreted as a spectral template and the corresponding temporal activation, respectively. $\boldsymbol{HU}$ can thus be viewed as the sum of the spectral templates scaled by time-varying amplitudes

2.3 Basic Properties of NMF

The number M of basis vectors is usually set smaller than the dimension K and the number N of data vectors. This is because when $M \geq K$ or $M \geq N$, there are trivial solutions to the factorization $\boldsymbol{Y} = \boldsymbol{HU}$. For example, when $M = K$, we have $\boldsymbol{Y} = \boldsymbol{IU}$ and when $M = N$, we have $\boldsymbol{Y} = \boldsymbol{HI}$, where $\boldsymbol{I}$ denotes an identity matrix. Obviously, neither of these decompositions provides information about the latent components underlying the data. When $M < \min(K, N)$, the factorization amounts to approximating the data matrix using a lower rank matrix, which provides meaningful information about the latent components. Geometrically, while PCA (singular value decomposition) tries to find a linear subspace to which observed vectors belong, NMF can be interpreted as finding a convex cone (see Fig. 2.2) that is closest to the entire set of observed vectors. The number M of basis vectors corresponds to the dimension of the convex cone, which depends on the data and is usually unknown. Thus, determining M is an important issue in NMF. Recent techniques for determining M will be mentioned in Sect. 2.8.

With NMF, the elements of the coefficient matrix $\boldsymbol{U}$ tend to become sparse as a side effect of the non-negativity constraint. The intuitive reason for this can be explained as follows. First, let us consider an unconstrained optimization problem $\hat{\boldsymbol{u}} = \operatorname{argmin}_{\boldsymbol{u}} \mathcal{D}(\boldsymbol{y}|\boldsymbol{Hu})$ where $\mathcal{D}(\cdot|\cdot)$ is a measure of the difference between two vectors. $\boldsymbol{H}\hat{\boldsymbol{u}}$ corresponds to the closest point from $\boldsymbol{y}$ in the subspace spanned by $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M$. If $\mathcal{D}$ is defined as an ℓ_2 norm, for example, this point simply corresponds to the orthogonal projection of $\boldsymbol{y}$ onto the subspace. Now, let us denote the solution to this optimization problem under the non-negativity constraint by $\tilde{\boldsymbol{u}}$. Except for a coincidental case where the unconstrained optimal solution $\hat{\boldsymbol{u}}$ satisfies the non-negativity constraint, $\boldsymbol{H}\tilde{\boldsymbol{u}}$ will be a closest point to $\hat{\boldsymbol{u}}$ in the convex cone shown in Fig. 2.2, namely some point on the boundary of the cone. This means at least one of the elements of the coefficient vector becomes 0. Therefore, the con-

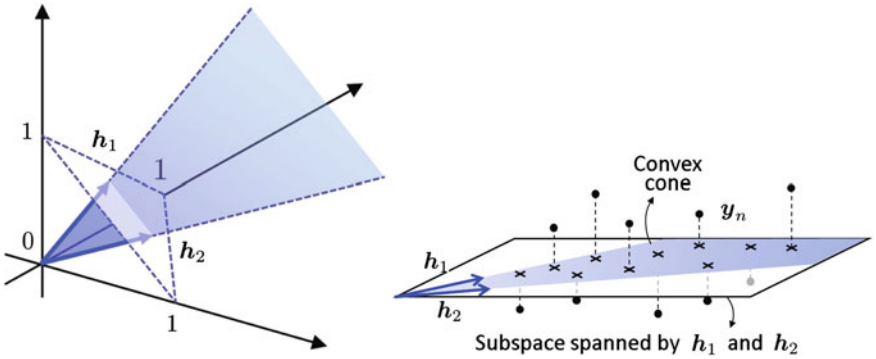


Fig. 2.2 Geometric understanding of NMF. Because of the non-negativity of $\boldsymbol{H}$, all basis vectors lie in the first quadrant. Because of the non-negativity of $\boldsymbol{U}$, $\boldsymbol{H}\boldsymbol{u}_n$ can only cover the area enclosed by the extended *lines* of all the basis vectors. Thus, NMF can be interpreted as finding a convex cone that is closest to the entire set of observed vectors

strained optimal solution $\tilde{\boldsymbol{u}}$ becomes relatively sparser (with a larger number of zero entries) than the unconstrained optimal solution $\hat{\boldsymbol{u}}$. This explains why NMF tends to produce sparse representations. It is important to note that sparsity is related to statistical independence (non-Gaussianity). Thus, roughly speaking, producing sparse representations implies that each row of the coefficient matrix tends to become uncorrelated. The above property also applies to the transposition of $\boldsymbol{Y} \simeq \boldsymbol{H}\boldsymbol{U}$, i.e., $\boldsymbol{Y}^\mathsf{T} \simeq \boldsymbol{U}^\mathsf{T}\boldsymbol{H}^\mathsf{T}$, meaning that $\boldsymbol{H}$ also tends to become sparse owing to the non-negativity constraint on $\boldsymbol{H}$.

2.4 NMF Algorithms

2.4.1 Positive Matrix Factorization and NMF

The original concept of NMF was first introduced by Paatero and Tapper in 1994 [2]. Within their formulation, they used the Frobenius norm of $\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{U}$ as a measure of the difference between $\boldsymbol{Y}$ and $\boldsymbol{H}\boldsymbol{U}$ and a logarithmic barrier function

$$B(\boldsymbol{H}, \boldsymbol{U}) = -\sum_{k,m} \log h_{k,m} - \sum_{m,n} \log u_{m,n} \tag{2.2}$$

as a penalizing term for violations of the non-negativity constraint, which approaches infinity as $h_{k,m}$ or $u_{m,n}$ approaches zero. They proposed a gradient-based optimization algorithm for minimizing the cost function defined as a weighted sum of these two terms.

Because of the property of the logarithmic barrier function, the elements of the matrices given by this method must always be positive (they never become zero). Thus, it is usually called "Positive Matrix Factorization (PMF)", which distinguishes it from NMF. Several years later, Lee and Seung proposed an iterative scheme called the multiplicative update algorithm, which ensures the non-negativity of $\boldsymbol{H}$ and $\boldsymbol{U}$ without using barrier functions [1]. Owing to the simplicity of its implementation, NMF has subsequently gained considerable momentum in a wide range of research areas.

2.4.2 Divergence Measures

NMF leads to different optimization problems according to the definition of the measure of the difference between $\boldsymbol{Y}$ and $\boldsymbol{HU}$. Lee and Seung have proposed deriving NMF algorithms using the Frobenius norm and the generalized Kullback-Leibler (KL) divergence (also known as the I divergence) [3] as the goodness-of-fit criteria. Of course, the optimal values of $\boldsymbol{H}$ and $\boldsymbol{U}$ depend on the choice of these criteria. It is desirable that the goodness-of-fit criterion be set according to the underlying generative process of the data $\boldsymbol{Y}$. For example, the Itakura-Saito (IS) divergence is often used as the model-fitting criterion for NMF when it is applied to power spectrum data [4, 5]. This is actually based on an assumption about the generative process of time-domain signals (as explained in Sect. 2.7.3).

For $y, x \in \mathbb{R}$, the Euclidean distance (squared error), the generalized KL divergence and the IS divergence of x from y are defined as

$$\mathcal{D}_{\mathrm{EU}}(y|x) = (y-x)^2, \tag{2.3}$$

$$\mathcal{D}_{\mathrm{KL}}(y|x) = y \log \frac{y}{x} - y + x, \tag{2.4}$$

$$\mathcal{D}_{\mathrm{IS}}(y|x) = \frac{y}{x} - \log \frac{y}{x} - 1, \tag{2.5}$$

respectively. All of these metrics become 0 only when $x = y$ and increase monotonically as x and y become more distant. Figure 2.3 shows the graph of each of these measures seen as a function of x. While the Euclidean distance is symmetric about $x = y$, the generalized KL divergence and the IS divergence are asymmetric and impose larger penalties when x is below y than when x is above y. It is also important to note that the IS divergence is invariant under the scaling of x and y since it is represented using only the ratio of x to y. By using these metrics, we can measure the difference between $\boldsymbol{HU}$ and $\boldsymbol{Y}$ with

$$D_{\cdot}(\boldsymbol{H}, \boldsymbol{U}) = \sum_{k,n} \mathcal{D}_{\cdot}\left(y_{k,n} \middle| \sum_{m} h_{k,m} u_{m,n}\right),$$

where $\cdot$ indicates EU, KL or IS.

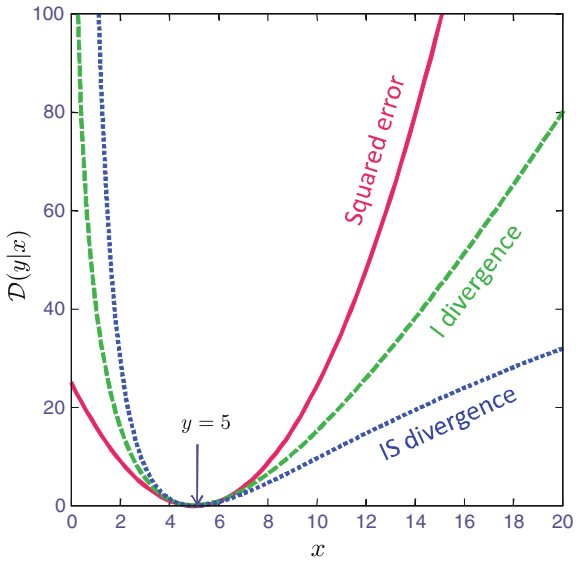


Fig. 2.3 Graph of $\mathcal{D}_{\mathrm{EU/KL/IS}}(y|x)$ as a function of x

2.4.3 Auxiliary Function Approach

The goal of NMF is to find optimal values for $\boldsymbol{H}$ and $\boldsymbol{U}$ that minimize one of these kinds of measures subject to the non-negativity constraint. Although it is usually difficult to obtain an analytical expression of the global optimum solution, one of the local optimum solutions can be searched for numerically using the "auxiliary function approach" (also known as the "Majorization-Minimization algorithm") [7, 25]. As explained later, the auxiliary function approach makes it possible to locally minimize an objective function by iteratively minimizing an auxiliary function whose lower bound is exactly equal to the objective function value. It should be noted that the Expectation-Maximization (EM) algorithm [8], a popular technique for maximum likelihood estimation from incomplete data, is a special case of this approach.

In NMF, the non-negativity constraint must be considered. If the objective function were given as the sum of individual terms, each relating to one matrix element, solving the constrained optimization problem would be relatively simple. But of course this is not the case. If we can use such a function as an auxiliary function, the constrained optimization problem of NMF can be solved in an iterative manner using the auxiliary function approach.

The definition of the auxiliary function and the principle of the auxiliary function approach are as follows.

Definition 2.1 Given an objective function $D(\theta)$ with the parameter $\theta = \{\theta_i\}_{1 \leq i \leq I}$, $G(\theta, \alpha)$ is defined as an *auxiliary function* of $D(\theta)$ if it satisfies

$$D(\theta) = \min_{\alpha} G(\theta, \alpha), \tag{2.6}$$

where we refer to α as *auxiliary variables*.

Theorem 2.1 *$D(\theta)$ is non-increasing under the updates:*

$$\alpha \leftarrow \operatorname*{argmin}_{\alpha} G(\theta, \alpha), \tag{2.7}$$

$$\theta_i \leftarrow \operatorname*{argmin}_{\theta_i} G(\theta, \alpha) \ \ (i = 1, \ldots, I). \tag{2.8}$$

Proof Let us set θ at an arbitrary value $\theta^{(\ell)}$ and define

$$\alpha^{(\ell+1)} = \operatorname*{argmin}_{\alpha} G(\theta^{(\ell)}, \alpha), \quad \theta^{(\ell+1)} = \left\{ \operatorname*{argmin}_{\theta_i} G(\theta, \alpha^{(\ell+1)}) \right\}_{1 \le i \le I}. \tag{2.9}$$

First, it is obvious that $D(\theta^{(\ell)}) = G(\theta^{(\ell)}, \alpha^{(\ell+1)})$. Next, we can confirm that $G(\theta^{(\ell)}, \alpha^{(\ell+1)}) \ge G(\theta^{(\ell+1)}, \alpha^{(\ell+1)})$. By definition, it is clear that $G(\theta^{(\ell+1)}, \alpha^{(\ell+1)}) \ge D(\theta^{(\ell+1)})$ and so we can finally show that $D(\theta^{(\ell)}) \ge D(\theta^{(\ell+1)})$. A sketch of this proof can be found in Fig. 2.4.

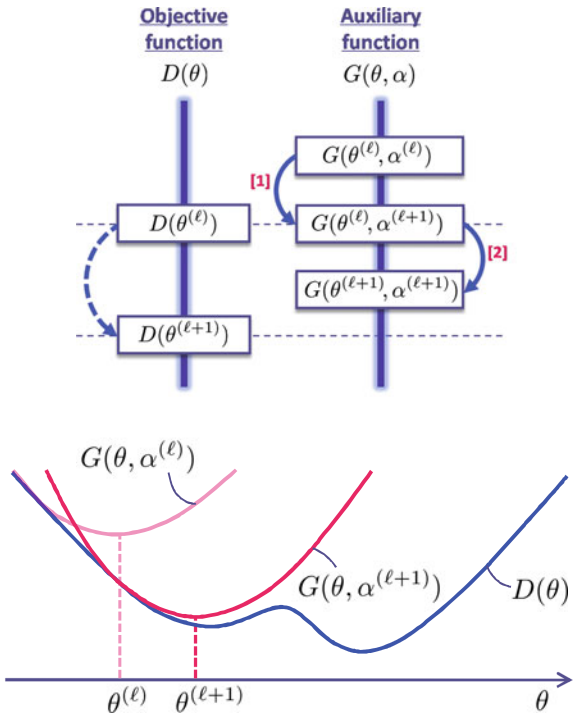


Fig. 2.4 Sketch of process of auxiliary function method

2.4.4 NMF Algorithm with Euclidean Distance

By employing the principle of the auxiliary function approach, we first derive an NMF algorithm using $D_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U})$ as the goodness-of-fit criterion. By using $\overset{z}{=}$ to denote equality up to a term independent of z, we can write $D_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U})$ as

$$D_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}) \overset{H,U}{=} \sum_{k,n} (-2y_{k,n}x_{k,n} + x_{k,n}^2), \tag{2.10}$$

where

$$x_{k,n} = \sum_m h_{k,m}u_{m,n}. \tag{2.11}$$

We want to design an auxiliary function such that the matrix elements are separated into individual terms. Note that $x_{k,n}^2$ is a term involving $h_{k,1}, \ldots, h_{k,M}$ and $u_{1,n}, \ldots, u_{M,n}$. Since a quadratic function is convex, we can employ Jensen's inequality to construct a desired auxiliary function.

Theorem 2.2 (Jensen's inequality for convex functions with non-negative arguments (Fig. 2.5)) *For an arbitrary convex function g with I non-negative arguments $z_1, \ldots, z_I$, we have*

$$g\Big(\sum_i z_i\Big) \leq \sum_i \lambda_i g\left(\frac{z_i}{\lambda_i}\right), \tag{2.12}$$

where $\lambda_1, \ldots, \lambda_1$ are non-negative weights satisfying $\sum_i \lambda_i = 1$. Equality in this inequality holds when

$$\lambda_i = \frac{z_i}{\sum_j z_j}. \tag{2.13}$$

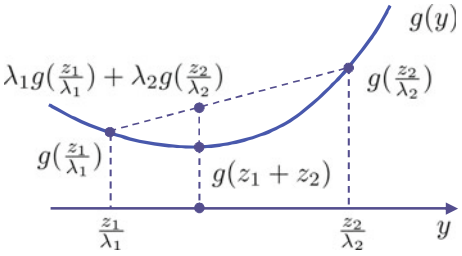


Fig. 2.5 Jensen's inequality for functions with non-negative arguments for $I = 2$ case

Since $h_{k,m}u_{m,n} \geq 0$, we can apply this to $x_{k,n}^2$

$$x_{k,n}^2 \leq \sum_m \lambda_{k,m,n} \left(\frac{h_{k,m}u_{m,n}}{\lambda_{k,m,n}} \right)^2, \tag{2.14}$$

where $\lambda_{k,m,n} \geq 0$, $\sum_m \lambda_{k,m,n} = 1$. Here, we notice that the right-hand side of this inequality is given as the sum of terms each relating to $h_{k,m}$ and $u_{m,n}$. It is also important to note that the equality holds when $\frac{h_{k,1}u_{1,n}}{\lambda_{k,1,n}} = \cdots = \frac{h_{k,M}u_{M,n}}{\lambda_{k,M,n}}$, namely

$$\lambda_{k,m,n} = \frac{h_{k,m}u_{m,n}}{x_{k,n}}. \tag{2.15}$$

Hence, the function obtained by replacing the term $x_{k,n}^2$ in $D_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U})$ with the right-hand side of Eq. (2.14)

$$G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda}) = \sum_{k,n} \left(y_{k,n}^2 - 2y_{k,n} \sum_m h_{k,m}u_{m,n} + \sum_m \frac{h_{k,m}^2 u_{m,n}^2}{\lambda_{k,m,n}} \right) \tag{2.16}$$

satisfies the requirement of an auxiliary function for $D_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U})$. Here, $\boldsymbol{\lambda} = \{\lambda_{k,m,n}\}_{K \times M \times N}$. By using $G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda})$, we can develop an iterative algorithm for locally minimizing $D_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U})$, that consists of performing

$$\boldsymbol{\lambda} \leftarrow \underset{\boldsymbol{\lambda}}{\operatorname{argmin}}\, G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda}), \tag{2.17}$$

$$\boldsymbol{H} \leftarrow \underset{\boldsymbol{H}}{\operatorname{argmin}}\, G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda}), \quad \boldsymbol{U} \leftarrow \underset{\boldsymbol{U}}{\operatorname{argmin}}\, G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda}). \tag{2.18}$$

First, Eq. (2.17) is given as Eq. (2.15) as mentioned above. Next, Eq. (2.18) must be solved subject to non-negativity. $G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda})$ is a quadratic function of each matrix element $h_{k,m}$, which can be minimized when

$$\hat{h}_{k,m} = \frac{\sum_n y_{k,n}u_{m,n}}{\sum_n u_{m,n}^2/\lambda_{k,m,n}}. \tag{2.19}$$

In the same way, $G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda})$ can be minimized with respect to $u_{m,n}$ when

$$\hat{u}_{m,n} = \frac{\sum_k y_{k,n}h_{k,m}}{\sum_k h_{k,m}^2/\lambda_{k,m,n}}. \tag{2.20}$$

If these values become negative, the minimizers of $G_{\mathrm{EU}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda})$ within the non-negativity constraint will obviously be $h_{k,m} = 0$ and $u_{m,n} = 0$. Thus, Eq. (2.18) is given as $h_{k,m} = \max\{\hat{h}_{k,m}, 0\}$ and $u_{m,n} = \max\{\hat{u}_{m,n}, 0\}$. Note, however, that when all the elements of $\boldsymbol{H}$, $\boldsymbol{U}$ and $\boldsymbol{\lambda}$ are non-negative, both (2.19) and (2.20) necessarily become non-negative. Hence, if the initial values of $\boldsymbol{H}$ and $\boldsymbol{U}$ are set at non-negative values, $h_{k,m}$ and $u_{m,n}$ will always be updated to non-negative values. In such a situation, the update equations for $h_{k,m}$ and $u_{m,n}$ can be written simply as $h_{k,m} = \hat{h}_{k,m}$ and $u_{m,n} = \hat{u}_{m,n}$. By substituting Eq. (2.17) into Eq. (2.18), we obtain the following algorithm:

NMF algorithm with the Euclidean distance

1. Set $\boldsymbol{H}$ and $\boldsymbol{U}$ at non-negative values.
2. Repeat the following updates until convergence.

$$h_{k,m} \leftarrow h_{k,m} \frac{\sum_n y_{k,n} u_{m,n}}{\sum_n x_{k,n} u_{m,n}} \qquad u_{m,n} \leftarrow u_{m,n} \frac{\sum_k y_{k,n} h_{k,m}}{\sum_k x_{k,n} h_{k,m}}$$

Since each variable is updated by multiplying the value at the previous iteration by a non-negative factor, this kind of algorithm is often referred to as a "multiplicative update algorithm" [1].

2.4.5 NMF Algorithm with I Divergence

An NMF algorithm using $D_{\mathrm{KL}}(\boldsymbol{H}, \boldsymbol{U})$ as a goodness-of-fit criterion can be derived in a similar way. $D_{\mathrm{KL}}(\boldsymbol{H}, \boldsymbol{U})$ is equal up to a constant term to

$$D_{\mathrm{KL}}(\boldsymbol{H}, \boldsymbol{U}) \overset{H,U}{=} \sum_{k,n} (-y_{k,n} \log x_{k,n} + x_{k,n}). \tag{2.21}$$

Here, $-y_{k,n} \log x_{k,n}$ is a nonlinear term involving $h_{k,1}, \ldots, h_{k,M}$ and $u_{1,n}, \ldots, u_{M,n}$. By using the fact that a negative logarithmic function is convex and $h_{k,m} u_{m,n} \geq 0$, we can apply Theorem 2.2

$$-\log x_{k,n} \leq -\sum_m \lambda_{k,m,n} \log \left(\frac{h_{k,m} u_{m,n}}{\lambda_{k,m,n}} \right)$$

to construct a desired auxiliary function, from which we obtain the following algorithm:

NMF algorithm with generalized KL divergence (*I* divergence)

1. Set $\boldsymbol{H}$ and $\boldsymbol{U}$ at non-negative values.
2. Repeat the following updates until convergence.

$$h_{k,m} \leftarrow h_{k,m} \frac{\sum_n y_{k,n} u_{m,n} / x_{k,n}}{\sum_n u_{m,n}} \qquad u_{m,n} \leftarrow u_{m,n} \frac{\sum_k y_{k,n} h_{k,m} / x_{k,n}}{\sum_k h_{k,m}}$$

2.4.6 NMF Algorithm with IS Divergence

Here, we show an NMF algorithm using the IS divergence as a goodness-of-fit criterion developed by the author in 2006 [9]. By omitting the terms that do not depend on $\boldsymbol{H}$ and $\boldsymbol{U}$, $D_{\mathrm{IS}}(\boldsymbol{H}, \boldsymbol{U})$ is written as

$$D_{\mathrm{IS}}(\boldsymbol{H}, \boldsymbol{U}) \stackrel{H,U}{=} \sum_{k,n} \left(\frac{y_{k,n}}{x_{k,n}} + \log x_{k,n} \right). \tag{2.22}$$

In a way similar to that described in the previous subsection, we want to design an auxiliary function such that the matrix elements are separated into individual terms. First, by using the fact that the reciprocal function is convex on a positive half-axis, $h_{k,m} u_{m,n} \geq 0$ and $y_{k,n} \geq 0$, we can apply Theorem 2.2 to the term $1/x_{k,n}$

$$\frac{1}{x_{k,n}} \leq \sum_m \lambda_{k,m,n} \left(1 \Big/ \frac{h_{k,m} u_{m,n}}{\lambda_{k,m,n}} \right), \tag{2.23}$$

where $\lambda_{k,m,n}$ is a positive weight satisfying $\lambda_{k,m,n} > 0$ and $\sum_m \lambda_{k,m,n} = 1$. Next, let us focus on the term $\log x_{k,n}$. Since the positive logarithmic function is concave (not convex), the strategy using Jensen's inequality cannot be used. However, we can apply a different inequality as described below. Given a differentiable concave function g, we can show that a tangent line to g at an arbitrary tangent point $\alpha \in \mathbb{R}$ lies entirely above the graph of g, namely for all $x \in \mathbb{R}$,

$$g(x) \leq g(\alpha) + (x - \alpha) g'(\alpha). \tag{2.24}$$

Obviously, the equality of this inequality holds if and only if $\alpha = x$. By applying this to $\log x_{k,n}$, we obtain

$$\log x_{k,n} \leq \log \alpha_{k,n} + \frac{1}{\alpha_{k,n}}(x_{k,n} - \alpha_{k,n}), \tag{2.25}$$

where $\alpha_{k,n}$ is an arbitrary real number. It is important to note that the right-hand side of this inequality is given as a first order function of the matrix elements. Hence, the function obtained by replacing the terms $1/x_{k,n}$ and $\log x_{k,n}$ in $D_{\mathrm{IS}}(\boldsymbol{H}, \boldsymbol{U})$ with the right-hand sides of Eqs. (2.23) and (2.25), such that

$$G_{\mathrm{IS}}(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\lambda}, \boldsymbol{\alpha}) = \sum_{k,n}\left(\sum_{m}\frac{y_{k,n}\lambda_{k,m,n}^2}{h_{k,m}u_{m,n}} + \sum_{m}\frac{h_{k,m}u_{m,n}}{\alpha_{k,n}} - \log y_{k,n} + \log \alpha_{k,n} - 2\right), \tag{2.26}$$

satisfies the requirement of an auxiliary function for $D_{\mathrm{IS}}(\boldsymbol{H}, \boldsymbol{U})$ [9]. Note that the equalities of Eqs. (2.23) and (2.25) hold if and only if

$$\lambda_{k,m,n} = \frac{h_{k,m}u_{m,n}}{x_{k,n}}, \quad \alpha_{k,n} = x_{k,n}. \tag{2.27}$$

By applying Theorem 2.1 and deriving each update equation, we obtain the following algorithm:

NMF algorithm with the IS divergence

1. Set $\boldsymbol{H}$ and $\boldsymbol{U}$ at non-negative values.
2. Repeat the following updates until convergence.

$$h_{k,m} \leftarrow h_{k,m}\left(\frac{\sum_{n} y_{k,n}u_{m,n}/x_{k,n}^2}{\sum_{n} u_{m,n}/x_{k,n}}\right)^{1/2} \qquad u_{m,n} \leftarrow u_{m,n}\left(\frac{\sum_{k} y_{k,n}h_{k,m}/x_{k,n}^2}{\sum_{k} h_{k,m}/x_{k,n}}\right)^{1/2}$$

2.4.7 NMF Algorithm with β Divergence

The three divergence measures given in Eqs. (2.3)–(2.5) can be described in a unified manner using a criterion called the β divergence [10]

$$\mathcal{D}_\beta(y|x) = y\frac{y^{\beta-1} - x^{\beta-1}}{\beta - 1} - \frac{y^\beta - x^\beta}{\beta}, \tag{2.28}$$

where β is a real number such that $\beta \neq 0$ and $\beta \neq 1$. By using the fact that $\lim_{\beta\to 0}(x^\beta - y^\beta)/\beta = \log(x/y)$, it can be confirmed that Eq. (2.28) reduces to the IS divergence when $\beta \to 0$, the I divergence when $\beta \to 1$ and the Euclidean distance when $\beta = 2$, respectively. Here, we show a generalized NMF algorithm using the β divergence as a goodness-of-fit criterion, that we have previously developed [11]. The first term $(y^{\beta-1} - x^{\beta-1})/(\beta - 1)$ of Eq. (2.28) is convex in x when $\beta \leq 2$ and is concave otherwise. On the other hand, the second term $-(y^\beta - x^\beta)/\beta$ is concave in x when $\beta \leq 1$ and is convex otherwise. In a way similar to the idea of [9], we can construct an auxiliary function by applying Eq. (2.12) to the convex term and Eq. (2.24) to the concave term. By using this auxiliary function, we can derive update equations given in closed form in the same way as in the previous subsections. The NMF algorithm derived using this idea is summarized as follows:

NMF algorithm with the β divergence

1. Set $\boldsymbol{H}$ and $\boldsymbol{U}$ at non-negative values, choose β and set $\varphi(\beta)$ at

$$\varphi(\beta) = \begin{cases} 1/(2-\beta) & (\beta < 1) \\ 1 & (1 \leq \beta \leq 2) \\ 1/(\beta - 1) & (\beta > 2) \end{cases}.$$

2. Repeat the following updates until convergence.

$$h_{k,m} \leftarrow h_{k,m}\left(\frac{\sum_n y_{k,n}x_{k,n}^{\beta-2}u_{m,n}}{\sum_n x_{k,n}^{\beta-1}u_{m,n}}\right)^{\varphi(\beta)} \qquad u_{m,n} \leftarrow u_{m,n}\left(\frac{\sum_k y_{k,n}x_{k,n}^{\beta-2}h_{k,m}}{\sum_k x_{k,n}^{\beta-1}h_{k,m}}\right)^{\varphi(\beta)}$$

It can be readily verified that the above algorithm reduces to the multiplicative update algorithms with the IS divergence, the I divergence and the Euclidean distance presented in Sects. 2.4.4, 2.4.5 and 2.4.6 when $\beta = 0, 1, 2$, respectively.

2.5 Interpretation of NMF as Generative Model

2.5.1 β Divergence Versus Tweedie Distribution

The optimization problems of NMF with the Euclidean distance, I divergence, IS divergence and β divergence are equivalent to the problems of the maximum

likelihood estimation of $\boldsymbol{H}$ and $\boldsymbol{U}$, where each element y_{kn} of $\boldsymbol{Y}$ is assumed to have been generated independently from the normal distribution, Poisson distribution, exponential distribution and Tweedie distribution with the mean $x_{k,n}$

$$y_{k,n} \sim \mathcal{N}(y_{k,n}; x_{k,n}, \sigma^2), \tag{2.29}$$
$$y_{k,n} \sim \text{Poisson}(y_{k,n}; x_{k,n}), \tag{2.30}$$
$$y_{k,n} \sim \text{Exponential}(y_{k,n}; x_{k,n}), \tag{2.31}$$
$$y_{k,n} \sim \text{Tweedie}(y_{k,n}; x_{k,n}, \phi), \tag{2.32}$$

respectively, where

$$\mathcal{N}(z; \mu, \sigma^2) = \tfrac{1}{\sqrt{2\pi\sigma^2}} e^{-(z-\mu)^2/2\sigma^2}, \tag{2.33}$$
$$\text{Poisson}(z; \mu) = \mu^z e^{-\mu}/z! \ \ (z \geq 0), \tag{2.34}$$
$$\text{Exponential}(z; \mu) = \tfrac{1}{\mu} e^{-z/\mu} \ \ (z \geq 0), \tag{2.35}$$
$$\text{Tweedie}(z; \mu, \phi) = a(z, \phi) e^{\frac{1}{\phi}(z\rho(\mu) - \kappa(\mu))}, \tag{2.36}$$
$$\rho(\mu) = \begin{cases} \frac{\mu^{\beta-1}-1}{\beta-1} & (\beta \neq 1) \\ \log \mu & (\beta = 1) \end{cases}, \ \kappa(\mu) = \begin{cases} \frac{\mu^{\beta}-1}{\beta} & (\beta \neq 0) \\ \log \mu & (\beta = 0). \end{cases}$$

This can be confirmed as follows. All the log-likelihoods $L(x_{k,n}) = \log p(y_{k,n}|x_{k,n})$ defined by Eqs. (2.29)–(2.32) are maximized when $x_{k,n} = y_{k,n}$. Thus, $L(y_{k,n}) \geq L(x_{k,n})$. Hence, the log-likelihood differences $L(y_{k,n}) - L(x_{k,n})$ can be regarded as non-negative measures of the dissimilarity between $y_{k,n}$ and $x_{k,n}$ that become 0 only when $x_{k,n} = y_{k,n}$. We can see that the log-likelihood differences $L(y_{k,n}) - L(x_{k,n})$ for Eqs. (2.29)–(2.32) are equal to Eqs. (2.3)–(2.5) and (2.28), respectively.

2.5.2 *Bregman Divergence Versus Natural Exponential Family*

As we have seen in the four examples above, an assumption regarding the divergence measure for a certain model-fitting problem is associated with a probability density function assumption regarding the observed data. In this subsection, I show that the class of probabilistic distributions belonging to the natural exponential family is associated with the class of goodness-of-fit criteria called the Bregman divergence and that the β divergence is a special case of the Bregman divergence. In the following, I will omit the subscripts k, n for simplicity and assume that an element y of the observed matrix follows a probability distribution belonging to the exponential family

$$y \sim \exp\left\{\eta T(y) - \psi(\eta) + c(y)\right\}, \tag{2.37}$$

where ψ is an infinitely differentiable, strictly convex function. η is called a natural parameter and is a function of the parameters characterizing the distribution. Here, we consider the case $T(y) = y$, whose distribution class is called the natural exponential family.

First, we introduce the Legendre transform of ψ

$$\phi(z) = \max_{v}(vz - \psi(v)). \tag{2.38}$$

Since ψ is a convex function, ϕ also becomes a convex function due to the property of the Legendre transform. By using v^* to denote $\phi(z)$, i.e., v that maximizes $vz - \psi(v)$, v^* satisfies $(vz - \psi(v))' = 0$, namely

$$\psi'(v^*) = z. \tag{2.39}$$

Next, by using the fact that the cumulant generating function of $y \sim \exp\{\eta y - \psi(\eta) + c(y)\}$ is given as $K(t) = \log \mathbb{E}[e^{yt}] = \psi(t + \eta) - \psi(\eta)$, we can write $x := \mathbb{E}[y] = K'(0)$ as

$$x = \psi'(\eta). \tag{2.40}$$

Since ψ is a convex function, ψ' is a one-to-one function. Thus, there is a one-to-one correspondence between η and x. By comparing Eq. (2.40) with Eq. (2.39), we can show that $\eta = \operatorname{argmax}_{v}(vx - \psi(v))$. $\phi(x)$ can thus be written as

$$\phi(x) = \eta(x)x - \psi(\eta(x)). \tag{2.41}$$

Note that here η is written as $\eta(x)$ to emphasize that it is a function of x. Here, by differentiating both sides of Eq. (2.41) with respect to x, we have

$$\phi'(x) = \eta(x) + \eta'(x)x - \psi'(\eta(x))\eta'(x). \tag{2.42}$$

By plugging Eq. (2.40) into Eq. (2.42), the second and third terms cancel each other out, thus resulting in $\phi'(x) = \eta(x)$.

By substituting the two relationships $\phi(x) = \eta x - \psi(\eta)$ and $\phi'(x) = \eta(x)$ given above into the probability density function of the natural exponential family $\exp\{\eta y - \psi(\eta) + c(y)\} = \exp\{\eta x - \psi(\eta) + \eta(y - x) + c(y)\}$, we obtain

$$p(y|x) = \exp\{\phi(x) + \phi'(x)(y - x) + c(y)\}. \tag{2.43}$$

Here, it is important to note that the log-likelihood of x, $L(x) = \log p(y|x) = \phi(x) + \phi'(x)(y - x) + c(y)$, is maximized when $x = y$, since $(\phi(x) + \phi'(x)(y - x))' = \phi''(x)(y - x)$. Thus, $L(y) \geq L(x)$. Hence, the log-likelihood difference $L(y) - L(x)$

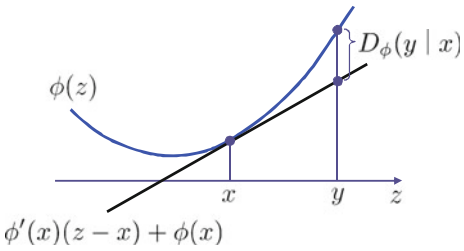


Fig. 2.6 Bregman divergence

$$\mathcal{D}_\phi(y|x) = \phi(y) - \phi(x) - \phi'(x)(y-x), \tag{2.44}$$

can be regarded as a non-negative measure of the dissimilarity between x and y that becomes 0 only when $x = y$. This measure is called the Bregman divergence [12]. As shown in Fig. 2.6, $D_\phi(y|x)$ corresponds to the difference between the convex function ϕ and its tangent line at point x. We can see from this figure that $D_\phi(y|x)$ is always non-negative and that $D_\phi(y|x)$ becomes 0 only when x and y are equal.

The β divergence introduced in Sect. 2.4.7 is a special case of the Bregman divergence with

$$\phi(x) = \begin{cases} -\log x + x - 1 & (\beta = 0) \\ x \log x - x + 1 & (\beta = 1) \\ \frac{x^\beta}{\beta(\beta-1)} - \frac{x}{\beta-1} + \frac{1}{\beta} & (\text{otherwise}) \end{cases} \quad [13]. \tag{2.45}$$

Thus, the Euclidean distance, I divergence, and IS divergence, which are special cases of the β divergence, are also special cases of the Bregman divergence.

An attempt was made by Dhillon and Sra to derive a multiplicative update algorithm for NMF with the Bregman divergence under a limited class of ϕ [14]. However, its generalization to an arbitrary ϕ has yet to be proposed.

2.6 Relation to Probabilistic Latent Semantic Analysis (pLSA)

The concept of probabilistic Latent Semantic Analysis (pLSA) [15], which is a technique that was originally developed for document clustering and indexing, is closely related to NMF. This section describes the relationship between these two techniques.

Let $y_{k,n}$ be the number of times word k occurs in document n. The histogram of all possible words $\boldsymbol{y}_n = (y_{1,n}, \ldots, y_{K,n})^\mathsf{T}$ in document n is called a document data. The number of times a particular word occurs may depend heavily on the topic of the document such as politics, economy, sports, entertainment, and culture. The aim of pLSA is to estimate topics from document data based on this dependence.

Let $p(k|m)$ be the probability that word k occurs when the topic is m and $p(m|n)$ be the probability that the topic of document n is m. Then, the probability $p(k|n)$ that word k occurs in document n can be written as

$$p(k|n) = \sum_m p(k|m)p(m|n). \tag{2.46}$$

By putting $x_{k,n} = p(k|n), h_{k,m} = p(k|m)$ and $u_{m,n} = p(m|n)$, and by arranging them in matrices $\boldsymbol{X} = (x_{k,n})_{K\times N}$, $\boldsymbol{H} = (h_{k,m})_{K\times M}$ and $\boldsymbol{U} = (u_{m,n})_{M\times N}$, Eq. (2.46) can be written in matrix notation as $\boldsymbol{X} = \boldsymbol{H}\boldsymbol{U}$. If each word in a set of document data is assumed to be generated independently according to the distribution $p(k|n)$, the probability that all the document data are generated becomes $\prod_{k,n} p(k|n)^{y_{k,n}}$. Since both $H_{k,m} = p(k|m)$ and $U_{m,n} = p(m|n)$ are unknown, the maximum likelihood estimation of $\boldsymbol{H}$ and $\boldsymbol{U}$ can be formulated as an optimization problem of maximizing

$$\log p(\boldsymbol{Y}|\boldsymbol{H}, \boldsymbol{U}) = \sum_{k,n} y_{k,n} \log x_{k,n}, \tag{2.47}$$

with respect to $\boldsymbol{H}$ and $\boldsymbol{U}$ subject to non-negativity and sum-to-one constraints: $h_{k,m} \geq 0$, $\sum_k h_{k,m} = 1$, $u_{m,n} \geq 0$, $\sum_m u_{m,n} = 1$. By comparing Eqs. (2.47) and (2.21), we notice that the above log-likelihood is exactly opposite to the first term of Eq. (2.21). Furthermore, as the second term of Eq. (2.21) can be seen as corresponding to a Lagrange multiplier term for $x_{k,n}$, the pLSA optimization problem has the same form as that of NMF with the I divergence criterion. Indeed, it turns out that the optimization algorithm described in Sect. 2.4.5 is equivalent to the expectation-maximization (EM) algorithm obtained by treating the topic index m as a latent variable up to the normalization of $\boldsymbol{H}$ and $\boldsymbol{U}$.

As described above, the way in which the likelihood function of pLSA is defined is different from NMF described in Sect. 2.5. While pLSA treats $h_{k,m}$ and $u_{m,n}$ as probability distributions over k and m, NMF treats them as random variables. Namely, pLSA is categorized as mixture models (models defined as the sum of probability distributions) whereas NMF is categorized as factor models (models defined as the distribution of the sum of random variables). The Bayesian extension of pLSA is called the latent Dirichlet allocation (LDA) [16] and the Bayesian extension of NMF with the I divergence criterion is discussed for example in [17].

2.7 Applications to Audio Signal Processing Problems

2.7.1 Audio Source Separation and Music Transcription

Smaragdis and Brown proposed an automatic music transcription method that uses NMF to decompose the magnitude (or power) spectrograms of music signals

into spectrograms associated with individual pitches [18]. With this approach, the magnitude (or power) spectrogram of a mixture signal, interpreted as a non-negative matrix $\boldsymbol{Y}$, is factorized into the product of two non-negative matrices $\boldsymbol{H}$ and $\boldsymbol{U}$ (See Fig. 2.1). This can in turn be interpreted as approximating the observed spectra at each time frame as a linear sum of basis spectra scaled by time-varying amplitudes, and amounts to decomposing the observed spectrogram into the sum of rank-1 spectrograms. As described in Sect. 2.3, an important feature of NMF is that its non-negativity constraint usually induces sparse representations, i.e., $\boldsymbol{U}$ with a relatively large number of zero entries. This means that each observed spectrum is parsimoniously represented using only a few active basis spectra. In such situations, the sequence of observed spectra can be approximated reasonably well when each basis spectrum expresses the spectrum of an underlying audio event that occurs frequently over the entire observed range. Thus, with music signals, each basis spectrum usually becomes the spectrum of a frequently used pitch in the music piece under analysis.

This approach is based on two assumptions; one is that magnitude (or power) spectra are additive and the other is that the magnitude spectrum of each sound source is constant up to the scale over time (i.e., only the scale of the spectrum is time-variant). However, these assumptions do not hold in reality. This section introduces some of the attempts that have been made to develop variants of NMF that aim to relax these assumptions.

2.7.2 Complex NMF

Audio signals in the time domain (sound waves) are additive. Since typical methods for time-frequency decomposition, such as the short-time Fourier transform (STFT) and the wavelet transform, are linear, complex spectrograms of audio signals are also additive. However, since the transformation of complex spectrograms into magnitude (or power) spectrograms is nonlinear, magnitude spectrograms are non-additive. Namely, the magnitude spectrum of the sum of two waveforms is not equal to the sum of the magnitude spectra of the two waveforms. This implies that decomposing a magnitude spectrogram into the sum of additive components does not necessarily lead to an appropriate decomposition of the audio signal.

To address this shortcoming of the NMF approach, I previously proposed a framework called the "Complex NMF" [19], which makes it possible to realize NMF-like signal decompositions in the complex spectrogram domain. The key idea behind the NMF approach was to model the magnitude spectrogram of a mixture signal as the sum of rank-1 magnitude spectrograms. By contrast, the key idea behind the proposed approach is to model the complex spectrogram of a mixture signal as the sum of complex spectrograms each having a rank-1 structure in the magnitude domain. This idea can be formulated as follows.

Let $a_{m,k,n} \in \mathbb{C}$ denote the complex spectrogram of source m. The complex spectrogram of a mixture signal consisting of M sources is given as

$$f_{k,n} = \sum_{m=1}^{M} a_{m,k,n} = \sum_{m} |a_{m,k,n}| e^{j\phi_{m,k,n}}, \tag{2.48}$$

where $\phi_{m,k,n}$ denotes the phase spectrogram of source m. Here, if we assume that the magnitude spectrogram of each source has a rank-1 structure, we can write $|a_{m,k,n}| = h_{k,m}u_{m,n}$. This leads to a complex spectrogram model of the form:

$$f_{k,n} = \sum_{m} h_{k,m}u_{m,n}e^{j\phi_{m,k,n}}. \tag{2.49}$$

It is important to emphasize that $\phi_{m,k,n}$ is indexed by n, meaning that this model allows the phase spectrum of each source to vary freely over time. The aim of Complex NMF is to fit this model to an observed complex spectrogram through the estimation of H, U and ϕ. It should be noted that unlike NMF, this model allows the components to cancel each other out (since the real and imaginary parts of the complex spectrum of each source can take either positive or negative values), and so when there are no constraints, it does not naturally produce sparse representations. Thus, to obtain sparse representations similar to NMF, some constraint is needed to induce the sparsity of $\boldsymbol{U}$. In [19], I formulated an optimization problem of minimizing

$$I(H, U, \phi) := \sum_{k,n} \left| y_{k,n} - f_{k,n} \right|^2 + 2\gamma \sum_{m,n} |u_{m,n}|^p, \tag{2.50}$$

with respect to H, U and ϕ where the second term is a sparse regularization term, and derived an iterative algorithm based on the auxiliary function approach. Here, $0 < p < 2$ and $\gamma \geq 0$ are constants. The main difficulty with this optimization problem lies in the nonlinear interdependence of $\phi_{1,k,n}, \ldots, \phi_{M,k,n}$ and the discontinuity of the gradients with respect to $u_{m,n}$. The nonlinear interdependence of $\phi_{1,k,n}, \ldots, \phi_{M,k,n}$ arises from the "square-of-sum" form in the first term of Eq. (2.50). To derive closed-form update equations using the auxiliary function approach in a similar way to Sect. 2.4.4, it is desirable to design an upper bound function that has a "sum-of-squares" form for this term. However, unlike Sect. 2.4.4, Theorem 2.2 cannot be applied in this case, since $h_{k,m}u_{m,n}e^{j\phi_{m,k,n}}$ is a complex number. Instead, in [19] I proposed invoking the following inequality:

Theorem 2.3 (Jensen's inequality for convex functions with complex arguments) *For an arbitrary convex function g with complex arguments y and $z_1, \ldots, z_I$, we have*

$$g\Big(y - \sum_{i} z_i\Big) \leq \sum_{i} \beta_i g\left(\frac{\alpha_i - z_i}{\beta_i}\right), \tag{2.51}$$

where $\alpha_1, \ldots, \alpha_1$ *are complex variables satisfying* $\sum_i \alpha = y$ *and* $\beta_1, \ldots, \beta_1$ *are positive weights satisfying* $\sum_i \beta = 1$. *Equality in this inequality holds when*

$$\alpha_i = z_i + \beta_i \Big(y - \sum_i z_i \Big). \tag{2.52}$$

Proof Since $\sum_i \alpha = y$, we can write $g(y - \sum_i z_i) = g(\sum_i (\alpha_i - z_i))$. By using arbitrary positive weights $\beta_1, \ldots, \beta_1$ that sum to one, we obtain

$$\begin{aligned} g\Big(\sum_i (\alpha_i - z_i) \Big) &= g\left(\sum_i \beta_i \frac{\alpha_i - z_i}{\beta_i} \right) \\ &\leq \sum_i \beta_i g\left(\frac{\alpha_i - z_i}{\beta_i} \right), \end{aligned} \tag{2.53}$$

where the second line follows from Jensen's inequality. Note that equality in this inequality holds when

$$\frac{\alpha_1 - z_1}{\beta_1} = \cdots = \frac{\alpha_I - z_I}{\beta_I}. \tag{2.54}$$

Letting Z denote the value of Eq. (2.54), α_i is given as $\alpha_i = z_i + \beta_i Z$. Since α_i must sum to y, i.e., $\sum_i \alpha_i = \sum_i z_i + Z = y$, Z is given by $Z = y - \sum_i z_i$. By substituting this into $\alpha_i = z_i + \beta_i Z$, we finally obtain Eq. (2.52).

As for the second term of Eq. (2.50), which is non-differentiable with respect to $u_{m,n}$, we can use the fact that, when $0 < p \leq 2$,

$$|u_{m,n}|^p \leq \frac{p|v_{m,n}|^{p-2}}{2} u_{m,n}^2 + \frac{2-p}{2} |v_{m,n}|^p, \tag{2.55}$$

to construct an upper bound function. Altogether, we obtain an auxiliary function

$$\begin{aligned} I^+(H, U, \phi, \alpha, V) := &\sum_{k,n,m} \frac{1}{\beta_{m,k,n}} \left| \alpha_{m,k,n} - h_{k,m} u_{m,n} e^{\mathrm{j}\phi_{m,k,n}} \right|^2 \\ &+ \gamma \sum_{m,n} \Big\{ p|v_{m,n}|^{p-2} {u_{m,n}}^2 + (2-p)|v_{m,n}|^p \Big\}, \end{aligned} \tag{2.56}$$

which has a "sum-of-squares" form. Here, $\beta_{m,k,n}$ is a positive weight that can be set arbitrarily subject to $\sum_m \beta_{m,k,n} = 1$. $\alpha_{m,k,n}$ and $v_{m,n}$ are auxiliary variables satisfying $\sum_m \alpha_{m,k,n} = y_{k,n}$. By using this, we can develop a convergence-guaranteed iterative algorithm with closed-form update equations.

2.7.3 Itakura-Saito NMF

Although the additivity of power spectra does not generally hold as mentioned above, it holds in the expectation sense if the signals are assumed to be samples drawn independently from stochastic processes.

If each underlying source signal in a mixture signal within a short-term segment is assumed to have been generated from a zero-mean circularly-stationary Gaussian process, each frequency component of the discrete Fourier transform of that segment independently follows a zero-mean complex normal distribution. Namely, if we let $s_{m,k,n}$ be a component of frequency k of source signal m within segment n (i.e., the complex spectrogram of source m), $s_{m,k,n}$ follows a zero-mean complex normal distribution

$$s_{m,k,n} \sim \mathcal{N}_{\mathbb{C}}\big(s_{m,k,n}; 0, \nu_{m,k,n}\big), \tag{2.57}$$

with variance $\nu_{m,k,n}$, where $\mathcal{N}_{\mathbb{C}}(z; \mu, \nu) = \frac{1}{\pi\nu} e^{-|z-\mu|^2/\nu}$. Note that $\nu_{m,k,n}$ corresponds to the expectation of the power spectrogram of source m, i.e., $\nu_{m,k,n} = \mathbb{E}[|s_{m,k,n}|^2]$. Now, if we assume that the complex spectrogram $y_{k,n}$ of an observed signal is given as $y_{k,n} = \sum_m s_{m,k,n}$, and that $s_{m,k,n}$ and $s_{m',k,n}$ $(m \neq m')$ are statistically independent, $y_{k,n}$ also follows a zero-mean complex normal distribution

$$y_{k,n} \sim \mathcal{N}_{\mathbb{C}}\Big(y_{k,n}; 0, \sum_m \nu_{m,k,n}\Big), \tag{2.58}$$

with variance $\sum_m \nu_{m,k,n}$. By putting $x_{k,n} = \sum_m \nu_{m,k,n}$, the log-likelihood of $x_{k,n}$ given an observation $y_{k,n}$ can be written as

$$L(x_{k,n}) = -\log \pi x_{k,n} - \frac{|y_{k,n}|^2}{x_{k,n}}. \tag{2.59}$$

Since this log-likelihood reaches maximum only when $x_{k,n} = |y_{k,n}|^2$, we have $L(|y_{k,n}|^2) \geq L(x_{k,n})$. We notice that the log-likelihood difference $L(|y_{k,n}|^2) - L(x_{k,n}) \geq 0$ is actually equal to the IS divergence between $|y_{k,n}|^2$ and $x_{k,n}$, i.e., $\mathcal{D}_{\mathrm{IS}}(|y_{k,n}|^2 | x_{k,n})$. Thus, if we assume the expectation of the power spectrogram of each source to have a rank-1 structure, i.e., $\nu_{m,k,n} = h_{k,m} u_{m,n}$, the maximum likelihood estimation of $\boldsymbol{H} = (h_{k,m})_{K \times M}$ and $\boldsymbol{U} = (u_{m,n})_{M \times N}$ is equivalent to the problem of NMF with the IS divergence criterion applied to the observed power spectrogram $|y_{k,n}|^2$ [4, 5].

2.7.4 NMF with Time-Varying Bases

When applying NMF to music spectrograms, we may expect the magnitude spectra of a single musical note produced by an instrument to be represented using a single basis spectrum scaled by time-varying amplitudes. However, its variations in time are actually much richer. For example, a piano note would be more accurately characterized by a succession of several basis spectra corresponding to, for example,

"attack," "decay," "sustain" and "release" segments. As another example, singing voices and string instruments have a particular musical effect, vibrato, which can be characterized by its "depth" (the range of pitch variation), and its "speed" (the rate at which the pitch varies). Several variants of NMF have been proposed to represent time-varying spectra by introducing the concept of time-varying bases [20–22].

One approach involves extending NMF to a convolutional version, which finds a decomposition of $\boldsymbol{Y}$ as

$$y_{k,n} \simeq x_{k,n} = \sum_m \sum_l h_{k,m,l} u_{m,n-l}, \tag{2.60}$$

where $h_{k,m,l}$ can be interpreted as the local time-frequency pattern of the mth audio event and $u_{m,n}$ represents its temporal activation. Since the problem at hand is to decompose the convolutive mixture, this approach is called "non-negative matrix factor deconvolution (NMFD)" [20].

NMFD assumes that the spectrum of each audio event evolves in time in exactly the same way every time it occurs. However, the speeds of the temporal variations are unlikely to be the same all the time. To cope with the varying speeds of the temporal variations of spectra, we proposed modeling the magnitude spectrogram of a mixture signal based on a factorial hidden Markov model (FHMM) formulation [22]. The idea is to model the spectrogram of a mixture signal as the sum of the outputs emitted from multiple HMMs, each representing the spectrogram of an underlying audio event (see Fig. 2.7). Thus, the problem is to find a decomposition of $\boldsymbol{Y}$ as

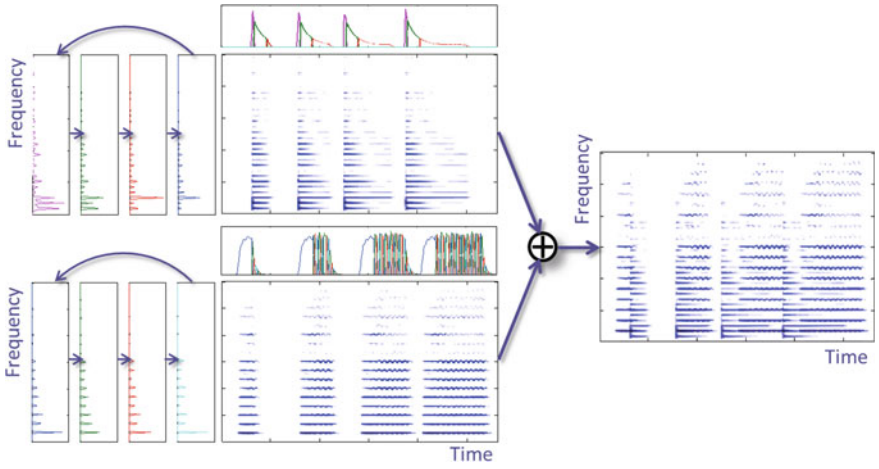


Fig. 2.7 Illustration of the factorial HMM approach [22]. The spectrogram of a mixture signal is modeled as the sum of the outputs emitted from multiple HMMs, each representing the spectrogram of an underlying audio event

$$y_{k,n} \simeq x_{k,n} = \sum_{m} h_{k,m}^{(z_{m,n})} u_{m,n}, \tag{2.61}$$

where $h_{k,m}^{(i)}$ denotes the basis spectrum at state i and $z_{m,n} \in \{1, \ldots, I_m\}$ denotes a state variable indicating which basis spectrum is activated at time n. The path of the state variables $z_{m,1}, \ldots, z_{m,N}$ is governed by a state transition probability $p(z_{m,n} = a | z_{m,n-1} = b) = \pi_{m,a,b}$.

2.7.5 Other NMF Variants

A number of constrained and regularized variants of NMF have been proposed specifically with the aim of solving audio source separation problems. Some examples are as follows. Virtanen proposed incorporating a temporal continuity constraint in the factorization process [23]. Raczyński proposed constraining each basis spectrum so that it had a harmonic structure [24]. Several groups (including mine) independently proposed combining the source-filter model with the NMF model [25–28]. I proposed incorporating a subprocess that involved clustering timbrally similar basis spectra in the factorization process [29].

2.7.6 Other Applications

NMF has found several interesting audio-related applications including speech enhancement [30], bandwidth extension [31], singing voice separation [32], drum sound extraction [33], formant tracking [35], echo cancellation [36], and the temporal decomposition of speech [37]. I proposed a blind dereverberation method inspired by the NMF algorithm in [27]. Multichannel extensions of NMF have been proposed independently by several groups (including mine) with an expectation that the modeling concept of NMF can also be useful for multichannel audio source separation problems [39–43].

2.8 Bayesian Nonparametric NMF

2.8.1 Determination of Basis Number

The determination of the number of bases is an important issue in NMF. Cemgil and Schmidt proposed formulating the problem of the basis number estimation for NMF as a model selection problem [17, 44]. By using $\boldsymbol{H}^{(M)}$ and $\boldsymbol{U}^{(M)}$ to denote the basis

and coefficient matrices with M bases, the marginal likelihood can be thought of as the likelihood function of M, since

$$p(\boldsymbol{Y}|M) = \iint p(\boldsymbol{Y}|\boldsymbol{H}^{(M)}, \boldsymbol{U}^{(M)}) p(\boldsymbol{H}^{(M)}|M) p(\boldsymbol{U}^{(M)}|M) \mathrm{d}\boldsymbol{H}^{(M)} \mathrm{d}\boldsymbol{U}^{(M)}. \quad (2.62)$$

As the exact marginal likelihood involves intractable integrals, some approximation of the log marginal likelihood is usually used as a criterion for model selection. The Bayesian Information Criterion (BIC) [45] and the variational Bayesian lower bound [46] are examples of such approximations. To determine the number of bases with model selection, we need to perform NMF with all possible M settings and find the best model structure by comparing the values of model selection criteria. Although this approach is indeed principled and well founded, such procedures can be time-consuming. By contrast, a framework called the Bayesian nonparameteric approach allows us to avoid performing an explicit model selection procedure and instead reduce this task to a single run of a parameter inference algorithm. In the following, we briefly show some examples of attempts that have been made to apply the Bayesian nonparameteric approach to NMF.

2.8.2 Beta Process NMF and Gamma Process NMF

A Bayesian nonparametric model is a Bayesian model on an infinite-dimensional parameter space. The Bayesian nonparameteric approach refers to a parameter inference framework for Bayesian nonparametric models, which makes it possible to infer the model complexity along with the model parameters by finding a minimal subset of parameters that can explain given observed data.

Bayesian nonparametric models (also known as infinite models) are typically described using stochastic processes. Up to now, many types of infinite models including infinite mixture models and infinite factor models have been proposed in the literature. For instance, infinite counterparts of mixture models, such as the Gaussian mixture model (GMM), hidden Markov model (HMM), probabilistic context-free grammar (PCFG), and probabilistic Latent Semantic Analysis (pLSA), can be constructed using a stochastic process called the Dirichlet process (DP) or its variants. While a Dirichlet distribution is a probabilistic distribution over a finite set of nonnegative numbers that sum to 1, the Dirichlet process can be thought of as an extension of it to an infinite set. Thus, the Dirichlet process is a generative model of a categorical distribution (probabilities of discrete random variables) with an infinite dimension, i.e., $\pi_1, \pi_2, \ldots \pi_\infty \in [0, 1]$ satisfying $\sum_{i=1}^{\infty} \pi_i = 1$, which can be used, for example, as a prior distribution over the mixture weights of mixture models. An important property of the Dirichlet process is its sparsity-inducing effect. The categorical distributions generated from a Dirichlet process tend to become sparse. Owing to this property, we can find a minimal subset of mixture components with non-zero weights that explains given observed data through parameter inference.

This is why the use of an infinite mixture model allows us to infer the adequate model complexity (the number of mixture components) from observed data. As mentioned in Sect. 2.6, pLSA can be understood as a particular case of NMF, where the number of mixture components (i.e., the latent topics) corresponds to the basis number. Thus, in a way similar to the NMF approach, it is technically possible to apply pLSA to a magnitude spectrogram by regarding it as document data, where the frequency and time indices are interpreted as the word and document indices, respectively [47], and an infinite counterpart of this approach can be constructed using a Dirichlet process [48]. On the other hand, infinite counterparts of factor models, such as NMF and Independent Component Analysis (ICA), can be constructed using stochastic processes called the beta process (BP) or gamma process (GP). Simply put, the beta process is a generative model of infinitely many variables within the range $[0, 1]$, $\pi_1, \pi_2, \ldots, \pi_\infty \in [0, 1]$, and the gamma process is a generative model of infinitely many non-negative variables, $\theta_1, \theta_2, \ldots, \theta_\infty \in [0, \infty)$. An infinite extension of NMF can be constructed using these stochastic processes. When using the beta process, we introduce a binary variable $z_{m,n} \in \{0, 1\}$ indicating whether the m-th basis exists in data n, with $z_{m,n} = 1$ if data n has a basis m and 0 otherwise. By using $z_{m,n}$, we define $x_{k,n}$ as $x_{k,n} = \sum_{m=1}^{\infty} z_{m,n} h_{k,m} u_{m,n}$ and place a beta process prior $\pi_{m,n} = p(z_{m,n} = 1)$ over $z_{1,n}, \ldots, z_{\infty,n}$ [49, 50]. An important feature of the beta process is its sparsity-inducing effect. The variables generated from a beta process tend to become sparse (most of the variables become almost 0). Owing to this property, we can find a minimal subset of bases that explains given observed data through parameter inference. When using the gamma process, we introduce a non-negative variable $\theta_m \in \mathbb{R}^{\geq 0}$ indicating the contribution made by basis m. By using θ_m, we define $x_{k,n}$ as $x_{k,n} = \sum_{m=1}^{\infty} \theta_m h_{k,m} u_{m,n}$, put some constraint on the scales of $h_{k,m}$ and $u_{m,n}$ (e.g., $\mathbb{E}[h_{k,m}] = 1$ and $\mathbb{E}[u_{m,n}] = 1$), and place a gamma process prior over $\theta_1, \ldots, \theta_\infty$ [22, 51]. An important feature of the gamma process is its sparsity-inducing effect as with the beta process. The variables generated from a gamma process tend to become sparse (most of the variables become almost 0). Owing to this property, we can find a minimal subset of bases that explains given observed data through parameter inference.

2.9 Summary

This chapter described some basic properties of NMF, effects induced by the non-negative constraints, how to derive an iterative algorithm for NMF, some attempts that have been made to apply NMF to audio processing problems, and extensions to the Bayesian nonparametric framework. Readers are referred to other review articles such as [52–55] for further details.

References

1. Lee, D. D., & Seung, H. S. (2000). Algorithms for nonnegative matrix factorization. In *Advances in NIPS* (pp. 556–562).
2. Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*, 111–126.
3. Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, *3*(1), 146–158.
4. Parry, R. M., & Essa, I. (2007). Phase-aware non-negative spectrogram factorization. In Proceedings of ICA (pp. 536–543).
5. Févotte, C., Bertin, N., & Durrieu, J.-L. (2009). Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural Computation*, *21*(3), 793–830.
6. Ortega, J. M., & Rheinboldt, W. C. (1970). *Iterative solutions of nonlinear equations in several variables*. New York: Academic Press.
7. Hunter, D. R., & Lange, K. (2000). Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, *9*, 60–77.
8. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, *39*(1), 1–38.
9. Kameoka, H., Goto, M., & Sagayama, S. (2006, August). Selective amplifier of periodic and non-periodic components in concurrent audio signals with spectral control envelopes. IPSJ Technical Report (vol. 2006-MUS-66, pp. 77–84) (in Japanese).
10. Eguchi, S., & Kano, Y. (2001). "Robustifying maximum likelihood estimation. Technical Report, Institute of Statistical Mathematics. Research Memo. 802.
11. Nakano, M., Kameoka, H., Le Roux, J., Kitano, Y., Ono, N., & Sagayama, S. (2010). Convergence-guaranteed multiplicative algorithms for non-negative matrix factorization with beta-divergence. In *Proceedings of MLSP* (pp. 283–288).
12. Bregman, L. M. (1967). The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, *7*(3), 210–217.
13. Hennequin, R., David, B., & Badeau, R. (2011). Beta-divergence as a subclass of Bregman divergence. *IEEE Signal Processing Letters*, *18*(2), 83–86.
14. Dhillon, I. S., & Sra, S. (2005). Generalized nonnegative matrix approximations with Bregman divergences. In *Advances in NIPS* (pp. 283–290).
15. Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of UAI* (pp. 289–296).
16. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022. (J. Lafferty (Ed.)).
17. Cemgil, A. T. (2008). Bayesian inference for nonnegative matrix factorization models, Technical Report CUED/F-INFENG/TR.609, University of Cambridge.
18. Smaragdis, P., & Brown, J. C. (2003). Non-negative matrix factorization for music transcription. In *Proceedings of WASPAA* (pp. 177–180).
19. Kameoka, H., Ono, N., Kashino, K., & Sagayama, S. (2009) Complex NMF: A new sparse representation for acoustic signals. In *Proceedings of ICASSP* (pp. 3437–3440).
20. Smaragdis, P. (2004). Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Proceedings of ICA* (pp. 494–499).
21. Ozerov, A. Févotte, C., & Charbit, M. (2009). Factorial scaled hidden Markov model for polyphonic audio representation and source separation. In *Proceedings of WASPAA* (pp. 121–124).
22. Nakano, M., Le Roux, J., Kameoka, H., Nakamura, T., Ono, N., & Sagayama, S. (2011). Bayesian nonparametric spectrogram modeling based on infinite factorial infinite hidden Markov model. In *Proceedings of WASPAA* (pp. 325–328).

23. Virtanen, T. (2007). Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing*, *15*(3), 1066–1074.
24. Raczynski, S. A., Ono, N., & Sagayama, S. (2007). Multipitch analisys with harmonic nonnegative matrix approximation. In *Proceedings of ISMIR* (pp. 381–386).
25. Virtanen, T., & Klapuri, A. (2006). Analysis of polyphonic audio using source-filter model and non-negative matrix factorization. In *Advances of NIPS*.
26. Vincent, E., Bertin, N., & Badeau, R. (2008) Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription. In *Proceedings of ICASSP* (pp. 109–112).
27. Kameoka, H., & Kashino, K. (2009). Composite autoregressive system for sparse source-filter representation of speech. In *Proceedings of ISCAS* (pp. 2477–2480).
28. Yoshii, K., & Goto, M. (2012, October). Infinite composite autoregressive models for music signal analysis. In *Proceedings of The 13th International Society for Music Information Retrieval Conference (ISMIR)* (pp. 79–84).
29. Kameoka, H., Nakano, M., Ochiai, K., Imoto, Y., Kashino, K., & Sagayama, S. (2012). Constrained and regularized variants of non-negative matrix factorization incorporating music-specific constraints. In *Proceedings of ICASSP* (pp. 5365–5368).
30. Smaragdis, P., Raj, B., & Shashanka, M. V. (2007). Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proceedings of ICA* (pp. 414–421).
31. Smaragdis, P., & Raj, B. (2007). Example-driven bandwidth expansion. In *Proceedings of WASPAA* (pp. 135–138).
32. Durrieu, J.-L., Richard, G., David, B., & Févotte, C. (2010). Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(3), 564–575.
33. Helén, M., & Virtanen, T. (2005). Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proceedings of EUSIPCO*.
34. Hurmalainen, A., Gemmeke, J., & Virtanen, T. (2011). Non-negative matrix deconvolution in noise robust speech recognition. In *Proceddings of ICASSP* (pp. 4588–4591).
35. Durrieu, J. -L., Thiran, J. -P. (2011). Sparse non-negative decomposition of speech power spectra for formant tracking. In *Proceedings of ICASSP* (pp. 5260–5263).
36. Togami, M., Kawaguchi, Y., Kokubo, H., & Obuchi, Y. (2010). Acoustic echo suppressor with multichannel semi-blind non-negative matrix factorization. In *Proceedings of APSIPA* (pp. 522–525).
37. Hiroya, S. (2013). Non-negative temporal decomposition of speech parameters by multiplicative update rules. *IEEE Transactions on Audio, Speech, and Language Processing*, *21*(10), 2108–2117.
38. Kameoka, H., Nakatani, T., & Yoshioka, T. (2009). Robust speech dereverberation based on non-negativity and sparse nature of speech spectrograms. In *Proceedings of ICASSP* (pp. 45–48).
39. Ozerov, A., & Févotte, C. (2010). Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(3), 550–563.
40. Kitano, Y., Kameoka, H., Izumi, Y., Ono, N., & Sagayama, S. (2010). A sparse component model of source sinals and its application to blind source separation. In *Proceedings of ICASSP* (pp. 4122–4125).
41. Sawada, H., Kameoka, H., Araki, S., & Ueda, N. (2011). New formulations and efficient algorithms for multichannel NMF. In *Proceedings of WASPAA* (pp. 153–156).
42. Sawada, H., Kameoka, H., Araki, S., & Ueda, N. (2012). Efficient algorithms for multichannel extensions of Itakura-Saito nonnegative matrix factorization. In *Proceedings of ICASSP* (pp. 261–264).
43. Higuchi, T., Takeda, H., Nakamura, T., Kameoka, H. (2014). A unified approach for underdetermined blind signal separation and source activity detection by multichannel factorial hidden Markov models. In *Proceedings of The 15th Annual Conference of the International Speech Communication Association (Interspeech 2014)* (pp. 850–854).

44. Schmidt, M. N., Winther, O., & Hansen, L. K. (2009). Bayesian non-negative matrix factorization. In *Proceedings of ICA* (pp. 540–547).
45. Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464.
46. Corduneanu, A., & Bishop, C. M. (2001). Variational Bayesian model selection for mixture distributions. In *Proceedings of AISTATS* (pp. 27–34).
47. Smaragdis, P., Raj, B., & Shashanka, M. (2006). A probabilistic latent variable model for acoustic modeling. In *Advances in NIPS*.
48. Yoshii, K., & Goto, M. (2012). A nonparametric Bayesian multipitch analyzer based on infinite latent harmonic allocation. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(3), 717–730.
49. Knowles, D., & Ghahramani, Z. (2007). Infinite sparse factor analysis and infinite independent components analysis.
50. Liang, D., Hoffman, M. D., & Ellis, D. P. W. (2013). Beta process sparse nonnegative matrix factorization for music.
51. Hoffman, M., Blei, D. & Cook, P. (2010). Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of ICML* (pp. 439–446).
52. Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. London: Wiley.
53. Kameoka, H. (2012). Non-negative matrix factorization with application to audio signal processing. *Acoustical Science and Technology*, *68*(11), 559–565. (in Japanese).
54. Sawada, H. (2012). Nonnegative matrix factorization and its applications to data/signal analysis. *IEICE Journal*, *95*, 829–833.
55. Smaragdis, P., Fevotte, C., Mysore, G., Mohammadiha, N., & Hoffman, M. (2014). Static and dynamic source separation using nonnegative factorizations: A unified view. In *IEEE Signal Processing Magazine* (pp. 66–75).

Chapter 3
Generalized Tensor PCA and Its Applications to Image Analysis

Kohei Inoue

Abstract Recently, principal component analysis (PCA) has made remarkable progress in some research areas including computer vision and pattern recognition. The objects handled by PCA have been extended from vectors to higher-order tensors, also known as multidimensional or N-way arrays including vectors (first order) and matrices (second order). A number of methods for tensor PCA have been proposed so far. In this chapter, we proposed a generalized formulation for dimensionality reduction of tensors and derive several state-of-the-art techniques for tensor PCA and low-rank approximation of tensors from the proposed general formulation. We also show experimental results of the applications of the derived methods to image analysis such as outliers removal, image compression, and face recognition.

Keywords Principal component analysis · Tensor · Dimensionality reduction · Simultaneous low-rank approximation of tensors · Image analysis

3.1 Introduction

Principal component analysis (PCA) [20] is one of the most fundamental techniques for multivariate data analysis. The original PCA is formulated for reducing the dimensions of vector data. Therefore, matrices and higher-order tensors have to be transformed into vectors for dimensionality reduction by PCA. Yang et al. [28] proposed two-dimensional PCA (2DPCA) for reducing the dimensions of matrices. 2DPCA directly reduces the number of columns of matrices without vectorization operation, while the number of rows is unchanged. Ye et al. [29] proposed generalized PCA (GPCA) which reduces the dimensions of both rows and columns of matrices. Cai et al. [4] and Ye [30] also proposed similar methods to GPCA. 2DPCA [28], GPCA [29], and its variants [4, 30] handle matrices or second-order tensors. Lu et al. [21] proposed a multilinear PCA (MPCA) framework for tensor object feature extraction. The MCPA framework includes the above PCA variants as the special cases. For example,

K. Inoue (✉)
Kyushu University, 4-9-1, Shiobaru, Minami-ku, Fukuoka, Japan
e-mail: k-inoue@design.kyushu-u.ac.jp

T. Sakata (ed.), *Applied Matrix and Tensor Variate Data Analysis*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55387-8_3

the relationship between MPCA [21], PCA [20], and 2DPCA [28] is discussed in [21]. Inoue and Urahama [17] generalized Ye's generalized low-rank approximation of matrices (GLRAM) [30] to that of higher-order tensors, and called it simultaneous low-rank approximation of tensors (SLRAT). De la Torre [11] proposed another unified least squares framework to formulate component analysis methods including PCA, linear discriminant analysis [14], canonical correlation analysis [16], locality preserving projections [2], and spectral clustering [23]. One drawback of those least squares estimation techniques based on Euclidean and Frobenius norms is the sensitivity to outliers. De la Torre and Black [13] developed a theory of robust subspace learning which includes robust PCA [12]. Inoue et al. [18] robustified MPCA [21] in two ways for two kinds of outliers: sample and intra-sample outliers [13]. Inoue et al. [19] also robustified SLRAT [17]. Candés et al. [5] proposed another robust PCA which minimizes a weighted combination of nuclear norm and l_1 norm to obtain a low-rank component and a sparse component of a data matrix.

In this chapter, we propose a generalized tensor PCA (GTPCA), which is a unified framework for tensor PCA including MPCA [21], robust MPCA [18], SLRAT [17], and robust SLRAT [19]. These tensor PCA variants are viewed as the special cases of GTPCA and applied to several image analysis tasks.

The rest of this chapter is organized as follows. Section 3.2 describes GTPCA, which generalizes several state-of-the-art techniques for tensor PCA. Section 3.3 derives the tensor PCA variants from GTPCA and summarizes the solution methods. Section 3.4 shows the experimental results of the applications of the derived tensor PCA variants to image analysis tasks. Finally, Sect. 3.5 concludes this chapter.

3.2 Generalized Tensor PCA

We use the minimal basic notation of tensor algebra necessary to describe the contents. The more detailed explanation of tensor algebra can be found in [6].

Let $\mathcal{A} = [a_{i_1 i_2 \ldots i_N}] \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth-order tensor or N-way array whose elements $a_{i_1 i_2 \ldots i_N}$ are indexed by $i_n \in \{1, 2, \ldots, I_n\}$ for $n = 1, 2, \ldots, N$, where $\mathbb{R}$ denotes the set of real numbers, and I_n is a positive integer. And let $X = [x_{j_n i_n}] \in \mathbb{R}^{J_n \times I_n}$ be a $J_n \times I_n$ matrix of which the (j_n, i_n) element is denoted by $x_{j_n i_n}$ for $j_n = 1, 2, \ldots, J_n$ and $i_n = 1, 2, \ldots, I_n$. Then, the n-mode product [9] of $\mathcal{A}$ and X is denoted by

$$\mathcal{A} \times_n X \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$$

of which the $(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N)$ element is given by

$$(\mathcal{A} \times_n X)_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} \equiv \sum_{i_n=1}^{I_n} a_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} x_{j_n i_n}.$$

Let $A_{(n)} \in \mathbb{R}^{I_n \times (I_{n+1} I_{n+2} \ldots I_N I_1 I_2 \ldots I_{n-1})}$ be the matrix unfolding [9] or the mode-n matricizing [1] of $\mathcal{A}$. Then $A_{(n)}$ contains the element $a_{i_1 i_2 \ldots i_N}$ of $\mathcal{A}$ at the position with row number i_n and column number equal to

$$(i_{n+1}-1)I_{n+2}I_{n+3}\ldots I_N I_1 I_2 \ldots I_{n-1} + (i_{n+2}-1)I_{n+3}I_{n+4}\ldots I_{n-1} + \cdots$$
$$+(i_N - 1)I_1 I_2 \ldots I_{n-1} + (i_1 - 1)I_2 I_3 \ldots I_{n-1} + (i_2 - 1)I_3 I_4 \ldots I_{n-1} + \cdots + i_{n-1}.$$

This notation enables us to convert the n-mode product into matrix multiplication:

$$(\mathcal{A} \times_n X)_{(n)} = X A_{(n)}.$$

Let $Y = \{Y^{(1)}, Y^{(2)}, \ldots, Y^{(N)}\}$ be a set of matrices such that $Y^{(n)} = [y^{(n)}_{j_n i_n}] \in \mathbb{R}^{J_n \times I_n}$ for $n = 1, 2, \ldots, N$. Then, the sequence of the n-mode products $\mathcal{A}$ and $Y^{(1)}, Y^{(2)}, \ldots, Y^{(N)}$ is denoted by

$$\mathcal{A} \times \{Y\} \equiv \mathcal{A} \times_1 Y^{(1)} \times_2 Y^{(2)} \cdots \times_N Y^{(N)} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}.$$

Related to this notation, the following notation is also useful [1]:

$$\mathcal{A} \times_{-n} \{Y\} \equiv \mathcal{A} \times_1 Y^{(1)} \cdots \times_{n-1} Y^{(n-1)} \times_{n+1} Y^{(n+1)} \cdots \times_N Y^{(N)}$$
$$\in \mathbb{R}^{J_1 \times \cdots \times J_{n-1} \times I_n \times J_{n+1} \times \cdots \times J_N}.$$

Let $\{\mathcal{A}_1, \ldots, \mathcal{A}_M\}$ be a set of tensors such that $\mathcal{A}_m = [a_{m i_1 \ldots i_N}] \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ for $m = 1, \ldots, M$, where $a_{m i_1 \ldots i_N}$ indexed by $i_n \in \{1, 2, \ldots, I_n\}$ for $n = 1, 2, \ldots, N$ are the elements of $\mathcal{A}_m$. Then, generalized tensor PCA (GTPCA) is formulated as follows:

$$\min_{\mathcal{S}, \{\mathcal{D}_m\}_{m=1}^M, U} \sum_{m=1}^{M} f\left(\mathcal{A}_m - \mathcal{S} - \mathcal{D}_m \times \{U\}\right) \tag{3.1}$$

$$\text{subj.to} \quad U^{(n)T} U^{(n)} = I_{J_n \times J_n}, \quad n = 1, 2, \ldots, N, \tag{3.2}$$

where $f : \mathbb{R}^{I_1 \times \cdots \times I_N} \to \mathbb{R}$, $\mathcal{S} = [s_{i_1 \ldots i_N}] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, $\mathcal{D}_m = [d_{m j_1 \ldots j_N}] \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ for $m = 1, 2, \ldots, M$ and $U = \{U^{(1)}, U^{(2)}, \ldots, U^{(N)}\}$ is a set of matrices such that $U^{(n)} = [u^{(n)}_{i_n j_n}] \in \mathbb{R}^{I_n \times J_n}$ for $n = 1, 2, \ldots, N$. The superscript ^{T} denotes the matrix transpose and $I_{J_n \times J_n}$ denotes the $J_n \times J_n$ identity matrix. The constraints in (3.2) have the element-wise expression as follows:

$$\sum_{i_1=1}^{I_n} u^{(n)}_{i_n j_n} u^{(n)}_{i_n j'_n} = \delta_{j_n j'_n} = \begin{cases} 1 \text{ if } & j_n = j'_n \\ 0 \text{ if } & j_n \neq j'_n \end{cases}, \quad n = 1, 2, \ldots, N,$$

where δ_{j_n,j'_n} denotes the Kronecker delta [8], which can be simply expressed as

$$\delta_{j_n,j'_n} = u^{(n)}_{i_n j_n} u^{(n)}_{i_n j'_n}$$

by using Einstein summation convention [8], which will also be used in the following derivations.

3.3 Derivation of Tensor PCA Variants

In this section, we derive four tensor PCA variants: multilinear PCA (MPCA) [21], robust MPCA [18], simultaneous low-rank approximation of tensors (SLRAT) [17], and robust SLRAT [19] from GTPCA, and summarize the solution methods.

3.3.1 Multilinear PCA (MPCA)

Multilinear PCA (MPCA) proposed by Lu et al. [21] can be derived from the above GTPCA (3.1) with (3.2) as follows.

Assume that $\mathcal{S}$ in (3.1) is the mean tensor $\bar{\mathcal{A}}$ of $\{\mathcal{A}_1, \ldots, \mathcal{A}_M\}$;

$$\mathcal{S} = \bar{\mathcal{A}} \equiv \frac{1}{M} \sum_{m=1}^{M} \mathcal{A}_m,$$

and the function f in (3.1) has the form $f(\mathcal{A}) = \|\mathcal{A}\|_F^2$ for a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, where $\|\cdot\|_F$ denotes the Frobenius norm [9] which is defined by

$$\|\mathcal{A}\|_F \equiv \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle},$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product [9] defined for two tensors $\mathcal{A} = [a_{i_1 \ldots i_N}] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\mathcal{B} = [b_{i_1 \ldots i_N}] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ by

$$\langle \mathcal{A}, \mathcal{B} \rangle \equiv \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} a_{i_1 \ldots i_N} b_{i_1 \ldots i_N} = a_{i_1 \ldots i_N} b_{i_1 \ldots i_N}, \tag{3.3}$$

which is a special case of the contracted product [1] of two tensors $\mathcal{E} = [e_{i_1 \ldots i_M j_1 \ldots j_N}] \in \mathbb{R}^{I_1 \times \cdots \times I_M \times J_1 \times \cdots \times J_N}$ and $\mathcal{F} = [f_{i_1 \ldots i_M k_1 \ldots k_P}] \in \mathbb{R}^{I_1 \times \cdots \times I_M \times K_1 \times \cdots \times K_P}$ given by

$$\langle \mathcal{E}, \mathcal{F} \rangle_{\{1,\ldots,M;1,\ldots,M\}} \equiv e_{i_1 \ldots i_M j_1 \ldots j_N} f_{i_1 \ldots i_M k_1 \ldots k_P} \in \mathbb{R}^{J_1 \times \cdots \times J_N \times K_1 \times \cdots \times K_P}. \tag{3.4}$$

Using this notation, we can express the scalar product (3.3) as follows:

$$\langle \mathcal{A}, \mathcal{B} \rangle = \langle \mathcal{A}, \mathcal{B} \rangle_{\{1,\ldots,N;1,\ldots,N\}}.$$

Then, the objective function in (3.1) can be written as follows:

$$E^{\mathrm{MPCA}} \equiv \sum_{m=1}^{M} \left\| \mathcal{A}_m - \bar{\mathcal{A}} - \mathcal{D}_m \times \{U\} \right\|_F^2 = \sum_{m=1}^{M} \left\| \tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\} \right\|_F^2, \tag{3.5}$$

where $\tilde{\mathcal{A}}_m = [\tilde{a}_{mi_1 \ldots i_N}] = \mathcal{A}_m - \bar{\mathcal{A}}$. The necessary condition for the optimality of $d_{mj_1 \ldots j_N}$ in $\mathcal{D}_m$ is given by

$$-\frac{1}{2} \frac{\partial E^{\mathrm{MPCA}}}{\partial d_{mj_1 \ldots j_N}} = \left(\tilde{a}_{mi_1 \ldots i_N} - d_{mj'_1 \ldots j'_N} u^{(1)}_{i_1 j'_1} \cdots u^{(N)}_{i_N j'_N} \right) u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N} = 0,$$

from which, it follows that

$$\begin{aligned}
&\left(\tilde{a}_{mi_1 \ldots i_N} - d_{mj'_1 \ldots j'_N} u^{(1)}_{i_1 j'_1} \cdots u^{(N)}_{i_N j'_N} \right) u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N} \\
&= \tilde{a}_{mi_1 \ldots i_N} u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N} - d_{mj'_1 \ldots j'_N} u^{(1)}_{i_1 j_1} u^{(1)}_{i_1 j'_1} \cdots u^{(N)}_{i_N j_N} u^{(N)}_{i_N j'_N} \\
&= \tilde{a}_{mi_1 \ldots i_N} u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N} - d_{mj'_1 \ldots j'_N} \delta_{j_1, j'_1} \cdots \delta_{j_N, j'_N} = 0,
\end{aligned}$$

that is,

$$d_{mi_1 \ldots i_N} = \tilde{a}_{mi_1 \ldots i_N} u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N}$$

or

$$\mathcal{D}_m = \tilde{\mathcal{B}}_m \equiv \tilde{\mathcal{A}}_m \times \left\{ U^T \right\}, \tag{3.6}$$

where $U^T = \{ U^{(1)^T}, \ldots, U^{(N)^T} \}$. Substituting (3.6) for (3.5), we have

$$\begin{aligned}
E^{\mathrm{MPCA}} &= \sum_{m=1}^{M} \left\| \tilde{\mathcal{A}}_m - \tilde{\mathcal{B}}_m \times \{U\} \right\|_F^2 = \sum_{m=1}^{M} \left\| \tilde{\mathcal{A}}_m - \left(\tilde{\mathcal{A}}_m \times \{U^T\} \right) \times \{U\} \right\|_F^2 \\
&= \sum_{m=1}^{M} \left[\tilde{a}_{mi_1 \ldots i_N} - \left(\tilde{a}_{mi'_1 \ldots i'_N} u^{(1)}_{i'_1 j_1} \cdots u^{(N)}_{i'_N j_N} \right) u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N} \right] \\
&\qquad \left[\tilde{a}_{mi_1 \ldots i_N} - \left(\tilde{a}_{mi''_1 \ldots i''_N} u^{(1)}_{i''_1 j'_1} \cdots u^{(N)}_{i''_N j'_N} \right) u^{(1)}_{i_1 j'_1} \cdots u^{(N)}_{i_N j'_N} \right] \\
&= \sum_{m=1}^{M} \left[\tilde{a}_{mi_1 \ldots i_N} \tilde{a}_{mi_1 \ldots i_N} - \tilde{a}_{mi_1 \ldots i_N} \left(\tilde{a}_{mi''_1 \ldots i''_N} u^{(1)}_{i''_1 j'_1} \cdots u^{(N)}_{i''_N j'_N} \right) u^{(1)}_{i_1 j'_1} \cdots u^{(N)}_{i_N j'_N} \right. \\
&\qquad \left. - \tilde{a}_{mi_1 \ldots i_N} \left(\tilde{a}_{mi'_1 \ldots i'_N} u^{(1)}_{i'_1 j_1} \cdots u^{(N)}_{i'_N j_N} \right) u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N} \right.
\end{aligned}$$

$$
\begin{aligned}
&+\left(\tilde{a}_{mi'_1\ldots i'_N} u^{(1)}_{i'_1 j_1}\cdots u^{(N)}_{i'_N j_N}\right)\left(\tilde{a}_{mi''_1\ldots i''_N} u^{(1)}_{i''_1 j'_1}\cdots u^{(N)}_{i''_N j'_N}\right) u^{(1)}_{i_1 j_1} u^{(1)}_{i_1 j'_1}\cdots u^{(N)}_{i_N j_N} u^{(N)}_{i_N j'_N}\Big] \\
&= \sum_{m=1}^{M}\Big[\tilde{a}_{mi_1\ldots i_N}\tilde{a}_{mi_1\ldots i_N} - \left(\tilde{a}_{mi_1\ldots i_N} u^{(1)}_{i_1 j'_1}\cdots u^{(N)}_{i_N j'_N}\right)\left(\tilde{a}_{mi''_1\ldots i''_N} u^{(1)}_{i''_1 j'_1}\cdots u^{(N)}_{i''_N j'_N}\right) \\
&\quad -\left(\tilde{a}_{mi_1\ldots i_N} u^{(1)}_{i_1 j_1}\cdots u^{(N)}_{i_N j_N}\right)\left(\tilde{a}_{mi'_1\ldots i'_N} u^{(1)}_{i'_1 j_1}\cdots u^{(N)}_{i'_N j_N}\right) \\
&\quad +\left(\tilde{a}_{mi'_1\ldots i'_N} u^{(1)}_{i'_1 j_1}\cdots u^{(N)}_{i'_N j_N}\right)\left(\tilde{a}_{mi''_1\ldots i''_N} u^{(1)}_{i''_1 j'_1}\cdots u^{(N)}_{i''_N j'_N}\right)\delta_{j_1 j'_1}\cdots\delta_{j_N j'_N}\Big] \\
&= \sum_{m=1}^{M}\Big[\tilde{a}_{mi_1\ldots i_N}\tilde{a}_{mi_1\ldots i_N} - \left(\tilde{a}_{mi_1\ldots i_N} u^{(1)}_{i_1 j_1}\cdots u^{(N)}_{i_N j_N}\right)\left(\tilde{a}_{mi'_1\ldots i'_N} u^{(1)}_{i'_1 j_1}\cdots u^{(N)}_{i'_N j_N}\right)\Big] \\
&= \sum_{m=1}^{M}\left(\left\|\tilde{\mathcal{A}}_m\right\|_F^2 - \left\|\tilde{\mathcal{A}}_m\times\{U^T\}\right\|_F^2\right) = \sum_{m=1}^{M}\left(\left\|\tilde{\mathcal{A}}_m\right\|_F^2 - \left\|\tilde{\mathcal{B}}_m\right\|_F^2\right),
\end{aligned}
$$

where $\Psi_{\mathcal{A}} \equiv \sum_{m=1}^{M}\|\tilde{\mathcal{A}}_m\|_F^2 = \sum_{m=1}^{M}\|\mathcal{A}_m - \bar{\mathcal{A}}\|_F^2$ denotes the total scatter of $\{\mathcal{A}_m\}_{m=1}^{M}$ and is a constant independent of U. Therefore, minimizing $E^{\mathrm{MPCA}} = \Psi_{\mathcal{A}} - \Psi_{\mathcal{B}}$ is equivalent to maximizing $\Psi_{\mathcal{B}} = \sum_{m=1}^{M}\|\tilde{\mathcal{B}}_m\|_F^2 = \sum_{m=1}^{M}\|(\mathcal{A}_m - \bar{\mathcal{A}})\times\{U^T\}\|_F^2 = \sum_{m=1}^{M}\|\mathcal{A}_m\times\{U^T\} - \bar{\mathcal{A}}\times\{U^T\}\|_F^2 = \sum_{m=1}^{M}\|\mathcal{B}_m - \bar{\mathcal{B}}\|_F^2$, where $\mathcal{B}_m = \mathcal{A}_m\times\{U^T\}$ and $\bar{\mathcal{B}} = \frac{1}{M}\sum_{m=1}^{M}\mathcal{B}_m = \frac{1}{M}\sum_{m=1}^{M}\mathcal{A}_m\times\{U^T\} = \bar{\mathcal{A}}\times\{U^T\}$. Consequently, the maximization problem of total scatter for MPCA is obtained by

$$\max_{U}\ \Psi_{\mathcal{B}}$$

with the constraints in (3.2). This problem can be solved by an iterative method similar to the higher-order orthogonal iteration [10].

3.3.2 Robust MPCA (RMPCA)

MPCA [21] described in Sect. 3.3.1 is not robust to outliers. In this subsection, we derive two types of robustification of MPCA (RMPCA) [18] from GTPCA and summarize the solution methods.

3.3.2.1 RMPCA for Sample Outliers

Assume that the function f in (3.1) has the form

$$f(\mathcal{A}) = \rho(\|\mathcal{A}\|_F) \tag{3.7}$$

where ρ is a ρ-function for robust M-estimation. For example, the ρ-function for Welsch estimator [15] is given by

$$\rho(x) = 1 - e^{-\alpha x^2}, \tag{3.8}$$

where $\alpha > 0$. Then, the objective function in (3.1) can be written as follows:

$$E^{\mathrm{RS}} \equiv M - \sum_{m=1}^{M} e^{-\alpha \|\mathcal{A}_m - \mathcal{S} - \mathcal{D}_m \times \{U\}\|_F^2}.$$

Since M is a constant, the problem (3.1) for (3.7) with (3.8) is equivalent to

$$\max_{\mathcal{S}, \{\mathcal{D}_m\}_{m=1}^M, U} \tilde{E}^{\mathrm{RS}}, \tag{3.9}$$

where

$$\tilde{E}^{\mathrm{RS}} \equiv M - E^{\mathrm{RS}} = \sum_{m=1}^{M} e^{-\alpha \|\mathcal{A}_m - \mathcal{S} - \mathcal{D}_m \times \{U\}\|_F^2}.$$

The solution method for RMPCA for sample outliers formulated in (3.9) is summarized as follows. The Lagrange function for (3.9) with (3.2) is given by

$$L^{\mathrm{RS}} \equiv \tilde{E}^{\mathrm{RS}} + \alpha \sum_{n=1}^{N} \mathrm{tr}\left[\Lambda^{(n)} \left(U^{(n)T} U^{(n)} - I_{J_n \times J_n}\right)\right],$$

where tr denotes the matrix trace [22] and $\Lambda^{(n)} \in \mathbb{R}^{J_n \times J_n}$ for $n = 1, \ldots, N$ is a symmetric matrix of which the elements are the Lagrange multipliers. Then, we have the following necessary conditions for optimality:

$$\frac{\partial L^{\mathrm{RS}}}{\partial \mathcal{S}} = 2\alpha \sum_{m=1}^{M} (\mathcal{C}_m - \mathcal{S})\, e^{-\alpha \|\mathcal{C}_m - \mathcal{S}\|_F^2} = O_{I_1 \times \cdots \times I_N}, \tag{3.10}$$

where $\mathcal{C}_m = \mathcal{A}_m - \mathcal{D}_m \times \{U\}$ and $O_{I_1 \times \cdots \times I_N} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is the Nth-order zero tensor of which all the elements are zero, and

$$\frac{\partial L^{\mathrm{RS}}}{\partial \mathcal{D}_m} = 2\alpha \left(\tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\}\right) \times \left\{U^T\right\} e^{-\alpha \left\|\tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\}\right\|_F^2} = O_{J_1 \times \cdots \times J_N} \tag{3.11}$$

for $m = 1, \ldots, M$, where $\tilde{\mathcal{A}}_m = \mathcal{A}_m - \mathcal{S}$, and

$$\begin{aligned} \frac{1}{2\alpha} \frac{\partial L^{\mathrm{RS}}}{\partial U^{(n)}} = & \sum_{m=1}^{M} \left[\tilde{A}_{m(n)} - U^{(n)} D_{m(n)}^{(-n)}\right] \left(D_{m(n)}^{(-n)}\right)^T e^{-\alpha \left\|\tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\}\right\|_F^2} \\ & + U^{(n)} \Lambda^{(n)} = O_{I_n \times J_n} \end{aligned} \tag{3.12}$$

for $n = 1, \ldots, N$, where $\tilde{A}_{m(n)}$ and $D^{(-n)}_{m(n)}$ are the mode-n matricizing [1] of $\tilde{\mathcal{A}}_m$ and $\mathcal{D}^{(-n)}_m = \mathcal{D}_m \times_{-n} \{U\}$, respectively, and $O_{I_n \times J_n}$ is the $I_n \times J_n$ matrix of which all the elements are zero. We also have

$$\frac{1}{\alpha}\frac{\partial L^{\mathrm{RS}}}{\partial \Lambda^{(n)}} = U^{(n)^T} U^{(n)} - I_{J_n \times J_n} = O_{J_n \times J_n} \tag{3.13}$$

for $n = 1, \ldots, N$.

From (3.10) and (3.11), we have

$$\mathcal{S} = \frac{\sum_{m=1}^{M} \mathcal{C}_m e^{-\alpha \|\mathcal{C}_m - \mathcal{S}\|_F^2}}{\sum_{m=1}^{M} e^{-\alpha \|\mathcal{C}_m - \mathcal{S}\|_F^2}} \tag{3.14}$$

and

$$\mathcal{D}_m = \tilde{\mathcal{A}}_m \times \left\{U^T\right\}, \tag{3.15}$$

respectively. From (3.12), we have

$$U^{(n)} = P^{(n)} \left(Q^{(n)} - \Lambda^{(n)}\right)^{-1}, \tag{3.16}$$

where

$$P^{(n)} = \sum_{m=1}^{M} \tilde{A}_{m(n)} \left(D^{(-n)}_{m(n)}\right)^T e^{-\alpha \left\| \tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\} \right\|_F^2},$$
$$Q^{(n)} = \sum_{m=1}^{M} D^{(-n)}_{m(n)} \left(D^{(-n)}_{m(n)}\right)^T e^{-\alpha \left\| \tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\} \right\|_F^2}.$$

By substituting (3.16) into (3.13), we find that

$$P^{(n)^T} P^{(n)} = \left(Q^{(n)} - \Lambda^{(n)}\right)^T \left(Q^{(n)} - \Lambda^{(n)}\right). \tag{3.17}$$

Let

$$P^{(n)^T} P^{(n)} = V^{(n)} \Sigma^{(n)} V^{(n)^T} \tag{3.18}$$

be a spectral decomposition [22] of $P^{(n)^T} P^{(n)}$, where $\Sigma^{(n)}$ is a diagonal matrix of which the diagonal elements are the eigenvalues of $P^{(n)^T} P^{(n)}$ and $V^{(n)}$ is an orthogonal matrix of which the columns are the corresponding eigenvectors of $P^{(n)^T} P^{(n)}$. Then, it follows from (3.17) and (3.18) that

$$\Lambda^{(n)} = Q^{(n)} - \Sigma^{(n)^{1/2}} V^{(n)^T}.$$

Substituting this into (3.16), we have

$$U^{(n)} = P^{(n)} V^{(n)} \Sigma^{(n)-1/2}. \tag{3.19}$$

Consequently, the problem (3.9) with (3.2) can be solved by iteratively updating $U^{(n)}$, $\mathcal{D}_m$, and $\mathcal{S}$ with (3.19), (3.15), and (3.14), respectively.

3.3.2.2 RMPCA for Intra-Sample Outliers

Assume that the function f in (3.1) has the form

$$f(\mathcal{A}) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \rho\left(a_{i_1 \ldots i_N}\right) \tag{3.20}$$

where $\mathcal{A} = [a_{i_1 \ldots i_N}] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. If we use the ρ-function in (3.8), then the objective function in (3.1) can be written as

$$E^{\mathrm{RI}} \equiv \sum_{m=1}^{M} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \rho\left(\hat{a}_{mi_1 \ldots i_N}\right) = \Theta - \sum_{m=1}^{M} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} e^{-\alpha \hat{a}^2_{mi_1 \ldots i_N}},$$

where $\hat{a}_{mi_1 \ldots i_N} = a_{mi_1 \ldots i_N} - s_{i_1 \ldots i_N} - d_{mj_1 \ldots j_N} u^{(1)}_{i_1 j_1} \cdots u^{(N)}_{i_N j_N}$ and $\Theta = M \prod_{n=1}^{N} I_n$. Since Θ is a constant, the problem (3.1) for (3.20) with (3.8) is equivalent to

$$\max_{\mathcal{S}, \{\mathcal{D}_m\}_{m=1}^{M}, U} \tilde{E}^{\mathrm{RI}}, \tag{3.21}$$

where

$$\tilde{E}^{\mathrm{RI}} \equiv \Theta - E^{\mathrm{RI}} = \sum_{m=1}^{M} \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} e^{-\alpha \hat{a}^2_{mi_1 \ldots i_N}}.$$

The solution method for RMPCA for intra-sample outliers formulated in (3.21) is summarized as follows. The Lagrange function for (3.21) with (3.2) is given by

$$L^{\mathrm{RI}} \equiv \tilde{E}^{\mathrm{RI}} + \alpha \sum_{n=1}^{N} \mathrm{tr}\left[\tilde{\Lambda}^{(n)} \left(U^{(n)T} U^{(n)} - I_{J_n \times J_n}\right)\right],$$

where $\tilde{\Lambda}^{(n)} \in \mathbb{R}^{J_n \times J_n}$ for $n = 1, \ldots, N$ is a symmetric matrix of which the elements are the Lagrange multipliers. Then, we have the following necessary conditions for optimality:

$$\frac{\partial L^{\mathrm{RI}}}{\partial \mathcal{S}} = 2\alpha \sum_{m=1}^{M} (\mathcal{C}_m - \mathcal{S}) \odot \mathcal{E}_m = O_{I_1 \times \cdots \times I_N}, \tag{3.22}$$

where $\mathcal{E}_m = [e^{-\alpha d^2_{mi_1 \dots i_N}}] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ for $m = 1, \dots, M$ and $\odot$ denotes the Hadamard product or element-wise product [22] of tensors, and

$$\frac{\partial L^{\mathrm{RI}}}{\partial \mathcal{D}_m} = 2\alpha \left[\left(\tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\} \right) \odot \mathcal{E}_m \right] \times \left\{ U^T \right\} = O_{J_1 \times \cdots \times J_N} \tag{3.23}$$

for $m = 1, \dots, M$, and

$$\frac{1}{2\alpha} \frac{\partial L^{\mathrm{RI}}}{\partial U^{(n)}} = \sum_{m=1}^{M} \left\langle \left(\tilde{\mathcal{A}}_m - \mathcal{D}_m \times \{U\} \right) \odot \mathcal{E}_m, \mathcal{D}_m^{(-n)} \right\rangle_{\{\Omega;\Omega\}} + U^{(n)} \tilde{\Lambda}^{(n)} = O_{I_n \times J_n} \tag{3.24}$$

for $n = 1, \dots, N$, where $\Omega = \{1, \dots, n-1, n+1, \dots, N\}$ and $\langle \cdot, \cdot \rangle_{\{\dots;\dots\}}$ denotes the contracted product [1] of tensors as described in (3.4), and

$$\frac{1}{\alpha} \frac{\partial \tilde{L}}{\partial \tilde{\Lambda}^{(n)}} = U^{(n)T} U^{(n)} - I_{J_n \times J_n} = O_{J_n \times J_n}$$

for $n = 1, \dots, N$.

From (3.22) and (3.23), we have

$$\mathcal{S} = \left(\sum_{m=1}^{M} \mathcal{C}_m \odot \mathcal{E}_m \right) \odot \mathcal{H},$$

where $\mathcal{H} = [1/\sum_{m=1}^{M} e^{-\alpha \hat{a}^2_{mi_1 \dots i_N}}] \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and (3.15), respectively. From (3.24), we have

$$U^{(n)} = \tilde{P}^{(n)} \left(\tilde{Q}^{(n)} - \tilde{\Lambda}^{(n)} \right)^{-1}, \tag{3.25}$$

where

$$\tilde{P}^{(n)} = \sum_{m=1}^{M} \left\langle \tilde{\mathcal{A}}_m \odot \mathcal{E}_m, \mathcal{D}_m^{(-n)} \right\rangle_{\{\Omega;\Omega\}},$$

$$\tilde{Q}^{(n)} = \sum_{m=1}^{M} \left\langle \mathcal{D}_m^{(-n)} \odot \mathcal{D}_m^{(-n)}, \mathcal{E}_m \right\rangle_{\{\Omega;\Omega\}}.$$

Since (3.25) is analogous to (3.16), we can derive the following equation in the same manner as the derivation of (3.19):

$$U^{(n)} = \tilde{P}^{(n)} \tilde{V}^{(n)} \left(\tilde{\Sigma}^{(n)} \right)^{-1/2},$$

where $\tilde{\Sigma}^{(n)}$ is a diagonal matrix of which the diagonal elements are the eigenvalues of $(\tilde{P}^{(n)})^T\tilde{P}^{(n)}$ and $\tilde{V}^{(n)}$ is an orthogonal matrix of which the columns are the corresponding eigenvectors of $(\tilde{P}^{(n)})^T\tilde{P}^{(n)}$.

3.3.3 *Simultaneous Low-Rank Approximation of Tensors (SLRAT)*

Assume that $\mathcal{S}$ in (3.1) is a zero tensor of which all the elements are zero, and the function f in (3.1) has the form $f(\mathcal{A}) = \|\mathcal{A}\|_F^2$ for a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times\cdots\times I_N}$. Then, the problem (3.1) becomes that of the simultaneous low-rank approximation of tensors $\{\mathcal{A}_m\}_{m=1}^M$ (SLRAT [17]) as follows:

$$\min_{\{\mathcal{D}_m\}_{m=1}^M, U} \sum_{m=1}^M \|\mathcal{A}_m - \mathcal{D}_m \times \{U\}\|_F^2 . \tag{3.26}$$

Let E^{SLRAT} be the objective function in (3.26), then, from $\partial E^{\mathrm{SLRAT}}/\partial \mathcal{D}_m = O_{J_1\times\cdots\times J_N}$, we have

$$\mathcal{D}_m = \mathcal{A}_m \times \left\{U^T\right\}. \tag{3.27}$$

Substituting this for E^{SLRAT}, we have

$$E^{\mathrm{SLRAT}} = \sum_{m=1}^M \|\mathcal{A}_m\|_F^2 - \sum_{m=1}^M \left\|\mathcal{A}_m \times \left\{U^T\right\}\right\|_F^2 .$$

Since the first term of the right side of this equation is a constant, the minimization problem (3.26) can be reduced to

$$\max_U \sum_{m=1}^M \left\|\mathcal{A}_m \times \left\{U^T\right\}\right\|_F^2 . \tag{3.28}$$

Let L^{SLRAT} be the Lagrange function for (3.28) with (3.2), then, from $\partial L^{\mathrm{SLRAT}}/\partial U^{(n)} = O_{I_n\times J_n}$, we have

$$\sum_{m=1}^M A_{m(n)}^{(-n)} A_{m(n)}^{(-n)^T} U^{(n)} = U^{(n)}\Lambda_n, \quad n = 1, \ldots, N, \tag{3.29}$$

where $A_{m(n)}^{(-n)}$ is the mode-n matricizing [1] of $\mathcal{A}_m \times_{-n} \{U^T\}$, and Λ_n is a diagonal matrix of which the diagonal elements are the Lagrange multipliers. The Eq. (3.29)

can be solved in an iterative manner as follows:

Algorithm 1.

Step 0: Initialize $U^{(n)}$ as $U^{(n,t)} = U^{(n,0)} = [I_{J_n \times J_n}, O_{J_n \times (I_n - J_n)}]^T \in \mathbb{R}^{I_n \times J_n}$ for $n = 1, \ldots, N$, where t denotes the number of iterations and is initialized as $t = 0$.

Step 1: Let $U_n^{(t)} = \{U^{(1,t+1)}, \ldots, U^{(n-1,t+1)}, U^{(n,t)}, \ldots, U^{(N,t)}\}$ for $t \geq 0$, and let $A_{m(n,t)}^{(-n)}$ be the mode-n matricizing [1] of $\mathcal{A}_m \times_{-n} \{U_n^{(t)^T}\}$. Then compute $U^{(n,t+1)}$ of which the columns are the eigenvectors of $\sum_{m=1}^{M} A_{m(n,t)}^{(-n)} A_{m(n,t)}^{(-n)^T}$ corresponding to the largest J_n eigenvalues for $n = 1, \ldots, N$.

Step 2: For $m = 1, \ldots, M$, compute $\mathcal{D}_m^{(t+1)}$ by (3.27) into which $U_N^{(t+1)}$ is substituted.

Step 3: Let $\mathrm{RMSE}^{(t+1)} = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \|\mathcal{A}_m - \mathcal{D}_m^{(t+1)} \times \{U_N^{(t+1)}\}\|_F^2}$ be the root-mean-squared error at the tth iteration. Note that the initial value of $\mathrm{RMSE}^{(t)}$ is given by $\mathrm{RMSE}^{(0)} = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \|\mathcal{A}_m - \mathcal{D}_m^{(0)} \times \{U_N^{(0)}\}\|_F^2}$. If, for $\epsilon > 0$, $[\mathrm{RMSE}^{(t)} - \mathrm{RMSE}^{(t+1)}]/\mathrm{RMSE}^{(t)} < \epsilon$, then stop, otherwise increase t by 1 and go to Step 1.

Instead of the initialization in Step 0, the full projection truncation [21] may also be used for initializing $U^{(n)}$.

For a fixed $U_n^{(t)}$, (3.29) is analytically solved for $U^{(n,t+1)}$ by the spectral decomposition [22] of $\sum_{m=1}^{M} A_{m(n,t)}^{(-n)} A_{m(n,t)}^{(-n)^T}$. Therefore, E^{SLRAT} monotonically decreases as t increases in Algorithm 1.

The generalized low-rank approximations of matrices (GLRAM) proposed by Ye [30] can be derived from SLRAT as follows: Let $N = 2$. Then, we have

$$\begin{aligned} E^{\mathrm{SLRAT}} &= \sum_{m=1}^{M} \left\| \mathcal{A}_m - \mathcal{D}_m \times_1 U^{(1)} \times_2 U^{(2)} \right\|_F^2 \\ &= \sum_{m=1}^{M} \left\| A_{m(1)} - U^{(1)} D_{m(1)} U^{(2)^T} \right\|_F^2, \end{aligned} \tag{3.30}$$

where $A_{m(1)}$ and $D_{m(1)}$ denote the mode-1 matricizing [1] of $\mathcal{A}_m$ and $\mathcal{D}_m$, respectively. The Eq. (3.30) coincides with the objective function of GLRAM [30].

3.3.4 Robust SLRAT

Assume that $\mathcal{S}$ in (3.1) is a zero tensor of which all the elements are zero, and the function f in (3.1) has the form $f(\mathcal{A}) = \rho(\|\mathcal{A}\|_F)$ as in (3.7), where ρ is the ρ-function for Welsch estimator [15] given by (3.8). Then, the objective function in (3.1) can be written as follows:

$$E^{\mathrm{RSLRAT}} \equiv M - \sum_{m=1}^{M} e^{-\alpha \|\mathcal{A}_m - \mathcal{D}_m \times \{U\}\|_F^2}, \tag{3.31}$$

which is the objective function of the robust version of SLRAT [19]. Since M is a constant, the minimization of (3.31) is equivalent to

$$\max_{U,\ \{\mathcal{D}_m\}_{m=1}^{M}} \tilde{E}^{\mathrm{RSLRAT}}, \tag{3.32}$$

where

$$\tilde{E}^{\mathrm{RSLRAT}} \equiv M - E^{\mathrm{RSLRAT}} = \sum_{m=1}^{M} e^{-\alpha \|\mathcal{A}_m - \mathcal{D}_m \times \{U\}\|_F^2}.$$

Then, it follows from $\partial \tilde{E}^{\mathrm{RSLRAT}} / \partial \mathcal{D}_m = O_{J_1 \times \cdots \times J_N}$ that

$$\mathcal{D}_m = \mathcal{A}_m \times \left\{U^T\right\}.$$

Also, we have

$$\frac{\partial \tilde{E}^{\mathrm{RSLRAT}}}{\partial U^{(n)}} = 2\alpha \sum_{m=1}^{M} \left(A_{m(n)} - U^{(n)} D_{m(n)}^{(-n)}\right) \left(D_{m(n)}^{(-n)}\right)^T e^{-\alpha \|\mathcal{A}_m - \mathcal{D}_m \times \{U\}\|_F^2},$$

where $A_{m(n)}$ denotes the mode-n matricizing [1] of $\mathcal{A}_m$. From $\partial \tilde{E}^{\mathrm{RSLRAT}} / \partial U^{(n)} = O_{I_n \times J_n}$, we have

$$U^{(n)} = \sum_{m=1}^{M} A_{m(n)} \left(D_{m(n)}^{(-n)}\right)^T e^{-\alpha \|\mathcal{A}_m - \mathcal{D}_m \times \{U\}\|_F^2} \left(\sum_{m=1}^{M} D_{m(n)}^{(-n)} \left(D_{m(n)}^{(-n)}\right)^T e^{-\alpha \|\mathcal{A}_m - \mathcal{D}_m \times \{U\}\|_F^2}\right)^{-1}. \tag{3.33}$$

Since the right side of (3.33) contains $U^{(n)}$ in $\{U\}$, we cannot solve (3.33) with respect to $U^{(n)}$ analytically. Instead, we solve (3.33) by an iterative method. First, we initialize $U^{(n)}$ as $U^{(n,0)} = [v_1^{(n)}, \ldots, v_{J_n}^{(n)}]$ where $v_1^{(n)}, \ldots, v_{J_n}^{(n)}$ are the eigenvectors of $\sum_{m=1}^{M} A_{m(n)} A_{m(n)}^T$ corresponding to the largest J_n eigenvalues. Next, we update $U^{(n)}$ as

$$\tilde{U}^{(n,t+1)} = \sum_{m=1}^{M} A_{m(n)} \left(D_{m(n,t)}^{(-n)}\right)^T e^{-\alpha \left\| \mathcal{A}_m - \hat{\mathcal{A}}_m^{(t)} \right\|_F^2} \left(\sum_{m=1}^{M} D_{m(n,t)}^{(-n)} \left(D_{m(n,t)}^{(-n)}\right)^T e^{-\alpha \left\| \mathcal{A}_m - \hat{\mathcal{A}}_m^{(t)} \right\|_F^2} \right)^{-1}, \tag{3.34}$$

where t is the number of iterations, $\hat{\mathcal{A}}_m^{(t)} = \mathcal{D}_m^{(t)} \times \{U^{(t)}\}$ for $\mathcal{D}_m^{(t)} = \mathcal{A}_m \times \{U^{(t)^T}\}$ and $U^{(t)} = \{U^{(1,t)}, \ldots, U^{(N,t)}\}$, and $D_{m(n,t)}^{(-n)}$ is the mode-n matricizing [1] of $\mathcal{D}_m^{(t)} \times_{-n} \{U^{(t)}\}$. Since (3.34) can be written as

$$\tilde{U}^{(n,t+1)} = U^{(n,t)} + \frac{1}{2\alpha} \frac{\partial \tilde{E}^{\mathrm{RSLRAT}(t)}}{\partial U^{(n,t)}} \left(\sum_{m=1}^{M} D_{m(n,t)}^{(-n)} \left(D_{m(n,t)}^{(-n)}\right)^T e^{-\alpha \left\| \mathcal{A}_m - \hat{\mathcal{A}}_m^{(t)} \right\|_F^2} \right)^{-1}, \tag{3.35}$$

this iterative algorithm can be interpreted as a gradient method [25]. Lastly, we orthogonalize $\tilde{U}^{(n,t+1)}$ as $U^{(n,t+1)} = \mathrm{orth}(\tilde{U}^{(n,t+1)})$, where $\mathrm{orth}(\cdot)$ is an orthogonalization function, in order to satisfy the constraints in (3.2). The above procedure is repeated until it converges.

3.4 Applications to Image Analysis

In this section, we apply the above tensor PCA variants to some image analysis tasks and show the experimental results on outliers removal, hyperspectral image compression, and face recognition.

3.4.1 Removing Outliers

In this subsection, we experimentally evaluate the performance of MPCA [21] and RMPCA [18], described in Sects. 3.3.1 and 3.3.2, on the ORL face image database [26]. The ORL database [26] contains face images of 40 persons. For each person, there are 10 different face images. That is, the total number of the images in the database is 400. The size of each image is 112×92 pixels, i.e., $I_1 = 112$, $I_2 = 92$, and $N = 2$. An example of face images in the database is shown in Fig. 3.1, where five images of a person are arranged along with an example of sample outliers (the rightmost image).

Reconstructed images with MPCA and RMPCA for sample outliers are shown in Fig. 3.2. Five reconstructed images from outlier-free data are shown in Fig. 3.2a, i.e., the clean reconstruction, where the rightmost image is the mean image of 10 face images of the person shown in Fig. 3.1. Reconstructed images with MPCA

Fig. 3.1 Input images: the first five images of the 27th person in the ORL database [26] and a sample outlier. © 2009 IEEE

Fig. 3.2 Reconstructed images and (*robust*) mean images: **a** outlier-free reconstruction by MPCA, **b** MPCA, **c** RMPCA. In each row, the left five images are the reconstructed ones, and the rightmost image is the mean one. © 2009 IEEE

and RMPCA for data including sample outliers are shown in Fig. 3.2b, c, respectively. In Fig. 3.2b, the left five reconstructed images and the rightmost mean image are disturbed by the sample outliers. On the other hand, in Fig. 3.2c, the left five reconstructed images are less sensitive to the outliers and the rightmost robust mean image is close to the outlier-free mean image in Fig. 3.2a. In this example, we set $J_1 = J_2 = 30$ for both MPCA and RMPCA, and $\alpha = 10^{-6}$ and $\epsilon = 10^{-6}$ for RMPCA. Although we selected the value of α manually in our experiments, some self-tuning algorithms [7, 27] might work well.

Let $\tilde{M}$ be the number of sample outliers per person. Then, we set $M = 10 + \tilde{M}$ for each person. Reconstructed images are computed as $\hat{\mathcal{A}}_m = \mathcal{D}_m \times \{U\} + \mathcal{S}$ for $m = 1, \ldots, 10$. Sample outliers are numbered from 11 to M. Reconstruction errors calculated with all face images in the ORL database [26] are shown in Fig. 3.3, in which we evaluated the errors by the root-mean-squared error (RMSE) defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M} \left\| \mathcal{A}_m - \hat{\mathcal{A}}_m \right\|_F^2}. \tag{3.36}$$

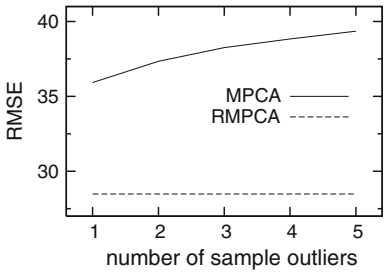


Fig. 3.3 Reconstruction errors. The *horizontal* and *vertical* axes denote the number of sample outliers per person and RMSE, respectively. *Solid* and *broken lines* denote MPCA and RMPCA, respectively. © 2009 IEEE

Fig. 3.4 Input images including intra-sample outliers. © 2009 IEEE

In this figure, the horizontal axis denotes the number of sample outliers per person and the vertical axis denotes the RMSE. MPCA and RMPCA are denoted by solid and broken lines, respectively. The RMSE for RMPCA is lower than that for MPCA. Although the RMSE for MPCA increases with the number of sample outliers, that for RMPCA is almost constant.

An example of face images including intra-sample outliers is shown in Fig. 3.4.

In each image, 2 % of pixels are outliers. In this experiment, we used the MATLAB function 'imnoise' with 'salt & pepper' option to generate intra-sample outliers. Reconstructed images with MPCA and RMPCA for intra-sample outliers are shown in Fig. 3.5a, b, respectively.

In Fig. 3.5a, the left five reconstructed images and the rightmost mean image are disturbed by the intra-sample outliers. On the other hand, in Fig. 3.5b, the left five reconstructed images and the rightmost robust mean image are less sensitive to

Fig. 3.5 Reconstructed images and (*robust*) mean images: **a** MPCA, **b** RMPCA. In each row, the left five images are the reconstructed ones, and the rightmost image is the mean one. © 2009 IEEE

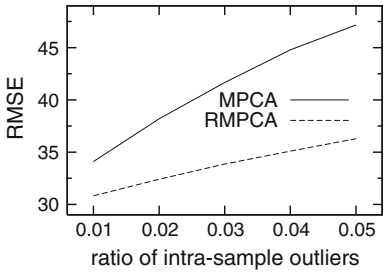


Fig. 3.6 Reconstruction errors. The *horizontal* and *vertical* axes denote the ratio of intra-sample outliers in all pixels and RMSE, respectively. *Solid* and *broken lines* denote MPCA and RMPCA, respectively.

the outliers. In this example, we set $\alpha = 10^{-3}$ and $M = 10$. Reconstruction errors evaluated by the RMSE in (3.36) are shown in Fig. 3.6, in which the horizontal axis denotes the ratio of intra-sample outliers in all pixels and the vertical axis denotes the RMSE. In this figure, MPCA and RMPCA are denoted by solid and broken lines, respectively. The RMSE for RMPCA is lower than that for MPCA. In contrast to the RMSE for RMPCA in Fig. 3.3, which is almost constant, that in Fig. 3.6 increases with the ratio of intra-sample outliers.

This result demonstrates the difficulty in detecting and removing the intra-sample outliers compared to the sample outliers.

3.4.2 Hyperspectral Image Compression

In this subsection, we show experimental results on Hyperspectral Image Compression with SLRAT [17] described in Sect. 3.3.3. We used a part of the Bear and Fruit Gray images in Hyperspectral Image Data provided by Brainard [3]. A hyperspectral image is a set of 31 monochromatic images corresponding to wavelengths between 400 and 700 nm in 10 nm steps. The provided image with 2020×2020 pixels is divided into $4 \times 4 = 16$ blocks each of which has 505×505 pixels. We clipped the top left 150×150 region from each block image and selected 30 images corresponding to wavelengths between 400 and 690 nm, that is, $I_1 = I_2 = 150$, $I_3 = 30$, and $M = 16$. We compared SLRAT with two kinds of GLRAM [30], i.e., the one reshapes a $150 \times 150 \times 30$ tensor into a 750×900 matrix by putting the 30 images of $I_1 \times I_2$ pixels into 5×6 blocks, and compresses 16 images of 750×900 pixels with GLRAM (GLRAM1). Another one deals with each $I_1 \times I_2$ image as a matrix and compresses $30 \times 16 = 480$ matrices with GLRAM (GLRAM2). The reconstruction errors are shown in Fig. 3.7, where the vertical axis denotes RMSE and the horizontal axis denotes the compression ratio defined by $[(I_1 + I_2 + I_3)p + Mp^3]/(I_1 I_2 I_3 M)$ where $p = J_1 = J_2 = J_3$, and we varied the value of p as $p = 10,\ 20, \ldots, 150$ for SLRAT, $p = 50,\ 100, \ldots, 750$ for GLRAM1, and $p = 10,\ 20, \ldots, 150$ for GLRAM2. The solid, broken, and dotted (with $+$) lines denote SLRAT, GLRAM1, and GLRAM2, respectively. CPU times for solving

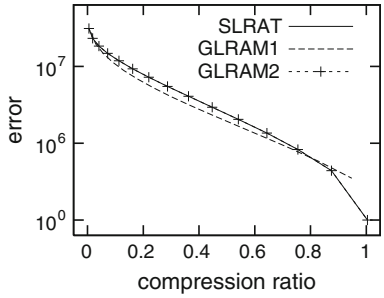


Fig. 3.7 Reconstruction errors. The *horizontal* and *vertical* axes denote the compression ratio and RMSE, respectively. *Solid*, *broken* and *dotted* (with +) lines denote SLRAT, GLRAM1 and GLRAM2, respectively. © 2006 IEICE

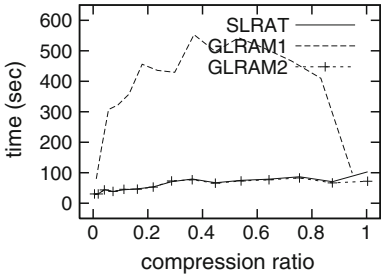


Fig. 3.8 CPU time. The *horizontal* and *vertical* axes denote the compression ratio and CPU time, respectively. *Solid*, *broken* and *dotted* (with +) lines denote SLRAT, GLRAM1 and GLRAM2, respectively. © 2006 IEICE

SLRAT, GLRAM1, and GLRAM2 problems are shown in Fig. 3.8. The performance of SLRAT is similar to that of GLRAM2.

3.4.3 Face Recognition

In this subsection, we apply SLRAT [17] and robust SLRAT [19] described in Sects. 3.3.3 and 3.3.4 to face recognition. The method for training and classification is summarized as follows.

Let $\mathcal{A}_{m_c} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ for $m_c = 1, \ldots, M_c$ and $c = 1, \ldots, C$ be a set of tensors for training, where C is the number of classes. Then, for each class c, we solve the following optimization problem:

$$\max_{U_c,\ \{\mathcal{D}_{m_c}\}_{m_c=1}^{M_c}} \sum_{m_c=1}^{M_c} e^{-\alpha \left\| \mathcal{A}_{m_c} - \mathcal{D}_{m_c} \times \{U_c\} \right\|_F^2} \tag{3.37}$$

$$\text{subj.to} \quad U_c^{(n)T} U_c^{(n)} = I_{J_n \times J_n}, \quad n = 1, \ldots, N, \tag{3.38}$$

where $U_c = \{U_c^{(1)}, \ldots, U_c^{(N)}\}$ is a set of $U_c^{(n)} \in \mathbb{R}^{I_n \times J_n}$ for $n = 1, \ldots, N$, and $\mathcal{D}_{m_c} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$. Let $U_c^* = \{U_{c*}^{(1)}, \ldots, U_{c*}^{(N)}\}$ be the solution of (3.37) with (3.38). Then, we store U_c^* for $c = 1, \ldots, C$ and use them for classifying test sets of tensors to be classified.

Let $\mathcal{A}_m \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ for $m = 1, \ldots, M$ be a set of tensors to be classified. Then, we solve the optimization problem in (3.32) with (3.2). Let $U^* = \{U_*^{(1)}, \ldots, U_*^{(N)}\}$ be the solution of the optimization problem. Then, we classify the set of tensors into the c^*th class selected by the following rule:

$$c^* = \arg \max_{c \in \{1, \ldots, C\}} S(U_c^*, U^*), \tag{3.39}$$

where $S(U_c^*, U^*)$ is a similarity between U_c^* and U^* and is defined as follows:

$$S(U_c^*, U^*) = \max_{\{p_{j_n}\}, \{q_{j_n}\}} \sum_{n=1}^{N} \sum_{j_n=1}^{J_n} p_{j_n}^T \left| U_{c*}^{(n)T} U_*^{(n)} \right|_{\text{abs}} q_{j_n} \tag{3.40}$$

$$\text{subj.to } p_{j_n}^T p_{j_n'} = \delta_{j_n, j_n'}, \quad q_{j_n}^T q_{j_n'} = \delta_{j_n, j_n'}, \tag{3.41}$$

$$p_{i_n j_n} \in \{0, 1\}, \quad q_{i_n j_n} \in \{0, 1\}, \tag{3.42}$$

where $p_{j_n} = [p_{i_n j_n}] \in \mathbb{R}^{J_n \times 1}$ and $q_{j_n} = [q_{i_n j_n}] \in \mathbb{R}^{J_n \times 1}$ for $i_n = 1, \ldots, J_n$ and $j_n = 1, \ldots, J_n$, and $|A|_{\text{abs}}$ denotes the absolute value of a matrix A [22]. The optimization problem in (3.40), (3.41), and (3.42) is no less than the assignment problem. We compute the optimal solution using Munkres algorithm [24].

An advantage of this method is that this method can accept various numbers of tensors in each set for training and testing, because the sizes of U_c and U are independent of M_c and M, respectively.

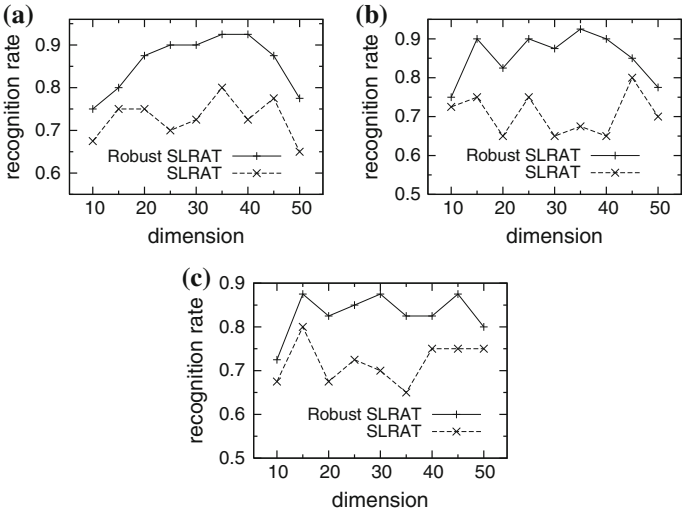


Fig. 3.9 Recognition rates: **a** $K = 1$, **b** $K = 2$, **c** $K = 3$. The *horizontal* and *vertical* axes denote the reduced dimension and the recognition rate, respectively. Solid lines with "+" marks and Broken lines with "×" marks denote robust SLRAT and SLRAT, respectively. © 2009 Springer

We next show the experimental results of face recognition by the above method. From the ORL face image database [26], we select the first five images per person for training and the remaining five images for testing. The number of classes coincides with that of persons in the database, i.e., $C = 40$. In the training phase, for each class c, we compute U_c^* for $c = 1, \ldots, C$ and store them. In the testing phase, in order to verify the robustness of robust SLRAT, we include noise images in the set of test images. For each person, we add $K = 1, 2, 3$ noise images and then compute U^*. Each set of test images with several noise images is classified on the basis of the classification rule described in (3.39). The recognition rates for $K = 1, 2$, and 3 are shown in Fig. 3.9a, b, and c, respectively. In each figure, the horizontal axis denotes the reduced dimension J_0 of each mode. We set J_1 and J_2 as $J_1 = J_2 = J_0$ for simplifying our experiments. The vertical axis denotes the recognition rate. The robust and the conventional SLRATs are denoted by solid lines with "+" marks and broken lines with "×" marks, respectively. The recognition rates for the robust SLRAT are higher than that for the conventional SLRAT.

3.5 Conclusion

In this chapter, we formulated a generalized tensor PCA (GTPCA) and derived multilinear PCA (MPCA), robust MPCA, simultaneous low-rank approximation of tensors (SLRAT), and robust SLRAT from GTPCA. We also showed the examples of applications of them to some image analysis tasks.

The derived tensor PCA variants optimize the corresponding objective functions including the Frobenius norm of tensors. In the proposed GTPCA formulation, other norms may also be used for other applications. Future work will include the developments of other tensor norm-based formulations and their optimizations for tensor variate data analysis.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number 23700212.

References

1. Bader, B. W., & Kolda, T. G. (2006). Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Transactions on Mathematical Software*, *32*, 635–653.
2. Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*, 1373–1396.
3. Brainard, D. H. (2014). Hyperspectral image data. http://color.psych.upenn.edu/hyperspectral/. (Cited 28 Oct 2014).
4. Cai, D., He, X., & Han, J. (2005). Subspace learning based on tensor analysis. Department of Computer Science Technical Report No. 2572, University of Illinois at Urbana-Champaign (UIUCDCS-R-2005-2572).
5. Candés, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, *58*(11), 1–37.

6. Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S. (2009). *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. New York: Wiley.
7. Comaniciu, D. (2003). An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*, 281–288.
8. Cubitt, T. (2014). Einstein summation convention and delta-functions. http://www.dr-qubit.org/teaching/summation_delta.pdf. (Cited 28 Oct 2014).
9. De Lathauwer, L., De Moor, B., & Vandewalle, J. (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, *21*, 1253–1278.
10. De Lathauwer, L., De Moor, B., & Vandewalle, J. (2000). On the best rank-1 and rank-(R1, R2, ..., RN) approximation of higher-order tensors. SIAM Journal on Matrix Analysis and Applications, 21, 1324–1342.
11. De la Torre, F. (2012). A least-squares framework for component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*, 1041–1055.
12. De la Torre, F., & Black, M. J. (2001). Robust principal component analysis for computer vision. *Proceedings of the ICCV*, 362–369.
13. De la Torre, F., & Black, M. J. (2003). A framework for robust subspace learning. *International Journal of Computer Vision*, *54*, 117–142.
14. Fisher, R. A. (1938). The statistical utilization of multiple measurements. *Annals of Eugenics*, *8*, 376–386.
15. Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (2005). *Robust Statistics: The approach based on influence functions (Wiley series in probability and statistics)*. New York: Wiley.
16. Hotelling, H. (1936). Relations between Two sets of variables. *Biometrika*, *28*, 321–377.
17. Inoue, K., & Urahama, K. (2006). Simultaneous low rank approximation of tensors. *Technical report of IEICE. PRMU*, *106*, 47–52.
18. Inoue, K., Hara, K., & Urahama, K. (2009). Robust multilinear principal component analysis. *Proceedings of the ICCV*, 591–597.
19. Inoue, K., Hara, K., & Urahama, K. (2009). Robust simultaneous low rank approximation of tensors. *Lecture Notes in Computer Science*, *5414*, 574–584.
20. Jolliffe, I. T. (2002). *Principal component analysis*. Springer Series in Statistics. 2nd ed.
21. Lu, H., Plataniotis, K. N., & Venetsanopoulos, A. N. (2008). MPCA: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, *19*, 18–39.
22. Lütkepohl, H. (1996). Handbook of matrices. Wiley.
23. Mohar, B. (1997). Some applications of laplace eigenvalues of graphs. *Graph Symmetry: Algebraic Methods and Applications*, 225–275.
24. Munkres, J. (1957). Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, *5*, 32–38.
25. Ortega, J. M., & Rheinboldt, W. G. (1970). *Iterative solution of nonlinear equations in several variables*. New York: Academic Press.
26. Samaria, F., & Harter, A. (1994). Parameterisation of a stochastic model for human face identification. In: *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*.
27. Zelnik-Manor, L., & Perona, P. (2005). Self-Tuning spectral clustering. *Proceedings of the NIPS*, 1601–1608.
28. Yang, J., Zhang, D., Frangi, A. F., & Yang, J. (2004). Two-Dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 131–137.
29. Ye, J., Janardan, R., & Li, Q. (2004). GPCA: An efficient dimension reduction scheme for image compression and retrieval. *Proceedings of the KDD*, 354–363.
30. Ye, J. (2005). Generalized low rank approximations of matrices. *Machine Learning*, *61*, 167–191.

Chapter 4
Matrix Factorization for Image Processing

Noboru Murata

Abstract Some of important methods for signal processing, such as principal component analysis (PCA), independent component analysis (ICA), non-negative matrix factorization (NMF), and sparse representation (SR), can be discussed in a unified framework where a data matrix is decomposed into a product of two specific matrices. Differences of those methods are understood as different constraints on decomposed matrices. Characteristics of those methods are discussed and compared by giving examples of image processing.

Keywords Principal component analysis · Independent component analysis · Non-negative matrix factorization · Sparse representation · Image processing

4.1 Introduction

In image processing aiming at such as noise reduction, restoration, and recognition, there are two important models of image generative process: probabilistic model and example-based model. Probabilistic models describe relationship among pixels of images in a stochastic way such as Markov random fields. With a small number of parameters, those models can capture intrinsic properties of real-world images, and they are applied on various image analysis such as texture modeling and image segmentation [23]. On the other hand, example-based models assume that images are represented by simple combinations of specific components or parts of objects. Those basic components can be extracted from a large amount of real-world images with appropriate learning methods.

Principal component analysis (PCA) [17] is one of the typical statistical tools for extracting common features from a set of observations and is used as an example-based model in image processing. PCA assumes that a datum is well approximated by a linear combination of orthogonal components. Since the procedure of

N. Murata (✉)
Faculty of Science and Engineering, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan
e-mail: noboru.murata@eb.waseda.ac.jp

T. Sakata (ed.), *Applied Matrix and Tensor Variate Data Analysis*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55387-8_4

estimating orthogonal components is reduced to the eigenvalue problem of the sample covariance matrix, it can be utilized in the analysis of a huge amount of data efficiently. For example, a method of "eigenfaces" [33] has been proposed in the field of face recognition, and many researchers have tried to improve and extend it.

Instead of simple computation, PCA does not always work properly because of its strong assumption on data. For example, let us consider the following two different processes of data generation:

- model A

$$\mathbf{x}_A = \begin{pmatrix} x_{A1} \\ x_{A2} \end{pmatrix} = \begin{pmatrix} 0.37 \\ 0.37 \end{pmatrix} z_1 + \begin{pmatrix} -0.11 \\ 0.11 \end{pmatrix} z_2,$$

- model B

$$\mathbf{x}_B = \begin{pmatrix} x_{B1} \\ x_{B2} \end{pmatrix} = \begin{pmatrix} 0.35 \\ 0.20 \end{pmatrix} z_1 + \begin{pmatrix} 0.20 \\ 0.35 \end{pmatrix} z_2.$$

Both models assume observed signal $\mathbf{x}$ is a linear combination of two components weighted by unseen source signals z_1 and z_2, and two components are orthogonal in model A, while not in model B. Suppose z_1 and z_2 are independently distributed with zero mean and unit variance. In this case, the distributions of $\mathbf{x}_A$ are different from that of $\mathbf{x}_B$ in general; however, the covariance matrices of $\mathbf{x}_A$ and $\mathbf{x}_B$ coincide. Particularly, when z_1 and z_2 are subject to standard normal distributions, the distributions of $\mathbf{x}_A$ and $\mathbf{x}_B$ themselves coincide, and we cannot distinguish that $\mathbf{x}$ comes from model A or model B. In general, PCA estimates model A even though $\mathbf{x}$ comes from model B because of the orthogonal assumption on the components.

For image analysis, this assumption of PCA is sometimes too strong; therefore, various example-based models for image processing have been proposed and discussed so far. In this chapter, we briefly summarize such models from a viewpoint of matrix factorization and then focus on a sparse representation method. The rest of this chapter is organized as follows. In Sect. 4.2, we give a unified framework of data representation with products of matrices and discuss specific constraints of data analysis methods such as PCA, ICA, NMF, and sparse representation. In Sect. 4.3, we discuss characteristics of the sparsity condition which is imposed on decomposed matrices. Particularly, we consider relation between sparsity and robustness of representation. In Sect. 4.4, we review typical algorithms for coefficient estimation and dictionary optimization for sparse matrix factorization. In Sect. 4.5, we present typical applications of sparse matrix factorization to image processing such as denoising, recognition, and super-resolution.

4.2 Data Representation by Matrix Factorization

In this section, we discuss several methods from a unified view of factorization of a data matrix, column vectors of which are multiple observations. First, we introduce some symbols and terms used in the following sections.

Let $\mathbf{x}$ be an observation which is an m-dimensional column vector:

$$\mathbf{x} = (x_1, \ldots, x_m)^{\mathrm{T}} \in \mathbb{R}^m.$$

When a dataset consists of n observations, they are indexed by a subscript as

$$\mathbf{x}_i, i = 1, \ldots, n.$$

Let X be a data matrix which is an expression of a dataset in matrix form:

$$X = (\mathbf{x}_1, \ldots, \mathbf{x}_n) \in \mathbb{R}^{m \times n} \tag{4.1}$$

We note that the above definition of X is often used in signal processing, but in statistics, X^{T} is defined as a data matrix instead.

We assume that observations are composed of common components and we call such a component *atom*. Atoms are denoted by m-dimensional vectors:

$$\mathbf{d} = (d_1, \ldots, d_m)^{\mathrm{T}} \in \mathbb{R}^m.$$

A set of atoms, which is indexed by a subscript, is called *dictionary*, and its matrix form is expressed as

$$D = (\mathbf{d}_1, \ldots, \mathbf{d}_k) \in \mathbb{R}^{m \times k}. \tag{4.2}$$

We assume that an observation is represented by a linear combination of atoms and combination coefficients are expressed with a k-dimensional vector as

$$\mathbf{z} = (z_1, \ldots, z_k)^{\mathrm{T}} \in \mathbb{R}^k.$$

We note that depending on the context of analysis, those coefficients are called by the name of unseen source signals, explanatory variables, hidden features, and so on. Coefficient vectors corresponding to observations are also indexed by the same subscript:

$$\mathbf{z}_i, i = 1, \ldots, n,$$

and a coefficient matrix is defined by a set of coefficient vectors in matrix form as

$$Z = (\mathbf{z}_1, \ldots, \mathbf{z}_n) \in \mathbb{R}^{k \times n}. \tag{4.3}$$

When we deal with rows of Z, row vectors are indexed by a superscript as

$$Z = (\mathbf{z}^1, \ldots, \mathbf{z}^k)^{\mathrm{T}} \in \mathbb{R}^{k \times n}. \tag{4.4}$$

Using those symbols, our problem is simply stated as follows: find good D and Z which approximate X by

$$X \simeq DZ.$$

Depending on constraints imposed on matrices D and Z, different multivariate analysis methods are derived:

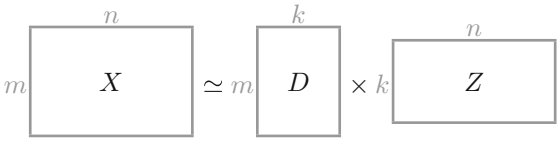
- Principal Component Analysis (PCA)
- Independent Component Analysis (ICA)
- Non-negative Matrix Factorization (NMF)
- Sparse Representation (SR)

In the following subsections, we discuss individual constraints.

4.2.1 Principal Component Analysis

Principal component analysis (PCA; also known as Karhunen-Loéve expansion) [17] is a method of decomposing a data matrix with linearly uncorrelated (orthogonal) atoms. At the request of orthogonality, the size of dictionary, i.e., the number of atoms, k is less than or equal to the dimension of data m. Each atom represents a direction of the corresponding principal component, and D is called *loading matrix*. Coefficients are called *scores* and scores for individual principal directions are also uncorrelated.

Factorization constraints for PCA are summarized as follows:



- D is full column rank ($m \geq k$),
- any column vectors $\mathbf{d}_i$ and $\mathbf{d}_j$ in D are unit and orthogonal,
- any row vectors $\mathbf{z}^i$ and $\mathbf{z}^j$ in Z are orthogonal.

The solution of PCA is given by the singular value decomposition of X. Let $U\Sigma V^{\mathrm{T}}$ be the singular value decomposition of X:

$$
\begin{aligned}
U &= (\mathbf{u}_1, \ldots, \mathbf{u}_m) \in \mathbb{R}^{m \times m}, \\
\Sigma &= \mathrm{diag}(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0) \in \mathbb{R}^{m \times n}, \\
V &= (\mathbf{v}_1, \ldots, \mathbf{v}_n) \in \mathbb{R}^{n \times n},
\end{aligned}
$$

where r is the rank of X. Consider the k-largest singular values, and let U_k, Σ_k and V_k be the sub-matrices of the corresponding singular values as

$$
\begin{aligned}
U_k &= (\mathbf{u}_1, \ldots, \mathbf{u}_k) \in \mathbb{R}^{m \times k}, \\
\Sigma_k &= \mathrm{diag}(\sigma_1, \ldots, \sigma_k) \in \mathbb{R}^{k \times k}, \\
V_k &= (\mathbf{v}_1, \ldots, \mathbf{v}_k) \in \mathbb{R}^{n \times k}.
\end{aligned}
$$

Then D and Z are given by

$$D = U_k, \quad Z = \Sigma_k V_k^{\mathrm{T}}. \tag{4.5}$$

An important variant of PCA is factor analysis (FA) [14]. In FA, D is also called *loading matrix*, and elements of Z is called *factors*. FA assumes that observations are contaminated by independent noises and that the structure of D is as simple as possible for the interpretation of analysis. Constraints of FA are summarized as

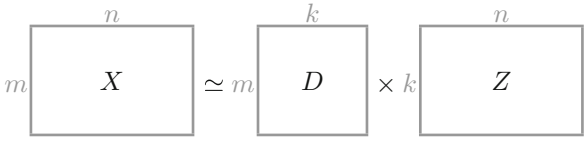
- D is full column rank ($m \geq k$),
- D has a simple structure,
- any row vectors $\mathbf{z}^i$ and $\mathbf{z}^j$ in Z are orthogonal.

Many kinds of criteria for structure simplicity of D are proposed (for more details, see [14]). In order to solve the problem under these constraints, iterative algorithms are usually employed.

4.2.2 Independent Component Analysis

Independent component analysis (ICA) [16] is a method for recovering statistically independent signals from observations. In ICA, observations are assumed to be linear mixtures of source signals, and mixing ratios correspond to atoms. This linear mixture assumption is similar to PCA, but ICA does not require the orthogonality of atoms. Each row vector of Z corresponds to n observations from a certain source, and a characteristic assumption of ICA is statistical independence of those sources. In most of ICA algorithms, D is assumed to be invertible, and objective functions are designed by estimating Z by $D^{-1}X$. Row vectors of Z are called *independent components*, and D and D^{-1} (if exists) are called *mixing matrix* and *de-mixing matrix*, respectively.

Constraints of ICA are summarized as follows.



- D is full column rank ($m \geq k$),
- row vectors $\mathbf{z}^i$ in Z are generated from independent distributions.

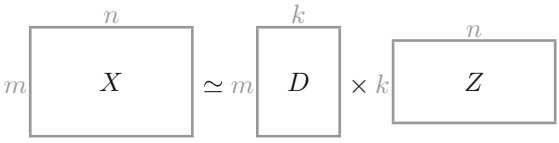
Since it is difficult to measure statistical independence empirically, various approximated independence measures are proposed such as nonlinear decorrelation of rows of Z and entropy minimization of Z. Moreover, constraints are highly

nonlinear in general; therefore, iterative algorithms and greedy algorithms are often utilized (for more details, see [16]).

4.2.3 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) [21, 22] is a method of approximating non-negative observations with products of non-negative atoms and coefficients. The idea of factorizing with positive matrices [28] is not new, but it becomes quite popular in image analysis after a fast algorithm based on multiplicative optimization has been proposed in [22]. In the optimization objective, there is no specific requirement for sparsity of D and Z, but the non-negative constraints on both of D and Z naturally make coefficients sparse, and also help interesting interpretation of estimated D and Z [21].

Constraints of NMF are simply summarized as follows.



- all the elements in D and Z are non-negative.

There are many variants of NMF. For example, l_1-norm regularization is applied in sparse NMF [15] because original NMF does not guarantee the sparsity of D and Z, the notion of Non-negative tensor factorization (NTF) is introduced in [31] in order to deal with array-structured data, and product rules of D and X are generalized in [13]. See [6] for recent advances.

4.2.4 Sparse Representation

Sparse representation (SR) or sparse coding [26, 27] is a method of representing observations with a linear combination of as few atoms as possible. For each observation, atoms are adaptively chosen from a sufficiently large size of dictionary, thus only a few coefficients have significant values and the rest are zero, i.e., coefficients are *sparse*. SR is inspired by visual processing of biological systems. In the visual system of animals, images acquired by the retina are once decomposed into simple components by receptive fields in visual cortex and again reconstructed to complex images hierarchically. In this process, reconstructed images are represented by as few components as possible in order to achieve robust recognition.

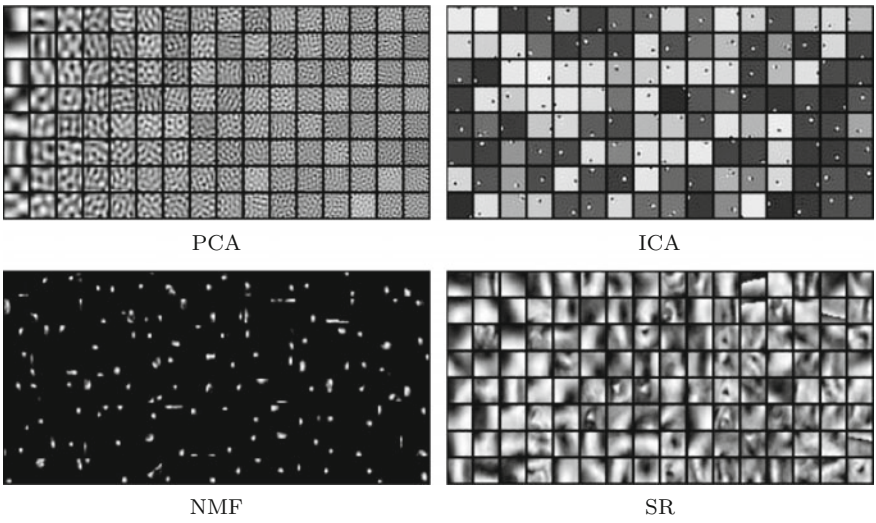
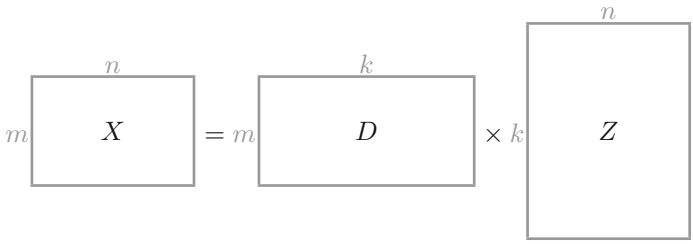


Fig. 4.1 Dictionaries learned from face images

Constraints of SR are summarized as follows.



- D is full row rank ($m \leq k$),
- column vectors $\mathbf{z}_i$ in Z are sparse.

As for example, dictionaries learned from face images (the FERET database [30]) by PCA, ICA, NMF, and SR are shown in Fig. 4.1. Figure 4.2 shows samples of face images.

In the following sections, we consider the decomposition under the SR constraints.

4.3 Characteristics of Sparseness

A general problem of sparse representation is written as follows: for a given data matrix X, find a dictionary D and a corresponding coefficient matrix Z as

$$\underset{D,Z}{\text{minimize}}\ R(Z) \quad \text{subject to} \quad C(X, D, Z) \leq \epsilon, \quad \epsilon > 0, \tag{4.6}$$

Fig. 4.2 Samples of face images from the FERET database

where C measures the cost (or loss) for approximating X by DZ and R measures or regularizes the goodness of Z. We note that in some cases specific conditions on D are imposed and the term $R(D, Z)$ is used instead of $R(Z)$. However, in this chapter, we only focus on the case that Z is regularized. Another formulation is given as

$$\underset{D,Z}{\text{minimize}}\, C(X, D, Z) \quad \text{subject to} \quad R(Z) \leq \tau, \quad \tau > 0, \tag{4.7}$$

where the roles of C and R are converted. By using the Lagrange multiplier method, those formulations are rewritten as

$$\underset{D,Z}{\text{minimize}}\, C(X, D, Z) + \lambda R(Z), \quad \lambda > 0. \tag{4.8}$$

Depending on the target problem, C and R should be carefully designed. In image processing, simple vector norms are commonly used for constructing C and R mainly because of computational simplicity. First, we introduce vector norms and related notation which we use in the following sections.

For a positive real number $p > 0$, the l_p-norm of a vector $\mathbf{z}$, $\|\mathbf{z}\|_p$, is defined as

$$\|\mathbf{z}\|_p^p = \sum_i |z_i|^p. \tag{4.9}$$

Special cases are $p = 2$ and $p = 1$ which are the Euclidean norm and the Manhattan norm, respectively. For $0 < p < 1$, $\|\cdot\|_p$ is not a proper norm in a mathematical sense, but we use the term "l_p-norm" even for $0 < p < 1$. We also define "l_0-norm" as follows. Let supp($\mathbf{z}$) be a index set of nonzero elements in a vector $\mathbf{z}$, i.e.,

$$\text{supp}(\mathbf{z}) = \{i | z_i \neq 0\}.$$

Then the l_0-norm of $\mathbf{z}$ is defined by

$$\|\mathbf{z}\|_0 = |\text{supp}(\mathbf{z})|, \tag{4.10}$$

where $|\cdot|$ denotes the cardinality of a set. In other words, the l_0-norm of $\mathbf{z}$ is the number of nonzero elements in $\mathbf{z}$.

A common choice of the cost C is the l_2-norm of the residual vectors, i.e.,

$$C(X, D, Z) = \|X - DZ\|_2^2 = \sum_{i=1}^{n} \|\mathbf{x}_i - D\mathbf{z}_i\|_2^2.$$

On the other hand, there are many possibilities for the choice of the regularization R. In order to control sparsity of $\mathbf{z}$, a natural choice might be the l_0-norm, i.e.,

$$R(Z) = \|Z\|_0 = \max_i \|\mathbf{z}_i\|_0.$$

However, strict optimization with the l_0-norm is known to be NP-hard [25]. To overcome this difficulty, several practical relaxations to the l_p-norm are proposed. The most popular choice is the l_1-norm, i.e.,

$$R(Z) = \|Z\|_1 = \max_i \|\mathbf{z}_i\|_1 \text{ or } \sum_{i=1}^{n} \|\mathbf{z}_i\|_1.$$

The relaxation of the l_0-norm to the l_1-norm is intensively investigated in terms of compressed sensing [8], and it is guaranteed that l_1-sparse representation can be appropriately estimated under mild conditions [3].

In the following subsections, we discuss the properties of sparsity from viewpoints of robustness and shrinkage estimation.

4.3.1 Robustness

Let us consider the following approximation problem in an m-dimensional space.

- We have a dictionary D of k ($> m$) atoms $\mathbf{d}_i$, $i = 1, \ldots, k$.
- We consider the subspaces spanned by τ atoms.

- One of the subspaces generates a datum, i.e., $D\mathbf{z}$, $\|\mathbf{z}\|_0 = \tau$.
- We estimate the target subspace from a contaminated observation $\mathbf{x} = D\mathbf{z} + \boldsymbol{\xi}$.
- What is the optimal size of τ?

Estimating the target subspace is equivalent to estimating which elements of $\mathbf{z}$ are nonzero from a observation $\mathbf{x}$.

In the first place, let us count the number of subspaces spanned by τ atoms out of k atoms. This number is given by

$$\binom{k}{\tau} = \frac{k!}{\tau!(k-\tau)!}, \tag{4.11}$$

which is equivalent to the number of k-dimensional $\{0, 1\}$-vectors whose τ elements are 1. As well known, this quantity is maximized at $\tau = k/2$ as shown in Fig. 4.3a.

Then, let us count the number of subspaces confused by noise $\boldsymbol{\xi}$. Such subspaces have overlapped atoms with the target subspace. As an example, let us consider subspaces with 50 %-overlap. This number is given by a sum over two quantities, i.e., more than half of τ atoms coincide and the rest of atoms are chosen from $k - \tau$ unused atoms.

$$\sum_{i=\tau/2}^{\tau} \binom{\tau}{i}\binom{k-\tau}{\tau-i}. \tag{4.12}$$

This quantity is also maximized at $\tau = k/2$ as shown in Fig. 4.3b.

From these calculations, we can consider the number of subspaces which are not affected by the contamination:

$$\begin{aligned}&\text{(the number of robust subspaces)}\\ &\quad = \frac{\text{(the number of subspaces)}}{\text{(the number of confusing subspaces)}} = \frac{\text{Eq. (4.11)}}{\text{Eq. (4.12)}}.\end{aligned}$$

This quantity is not maximized at $\tau = k/2$ as shown in Fig. 4.3c. When the observation is contaminated by smaller noise, the number of confusing subspaces becomes

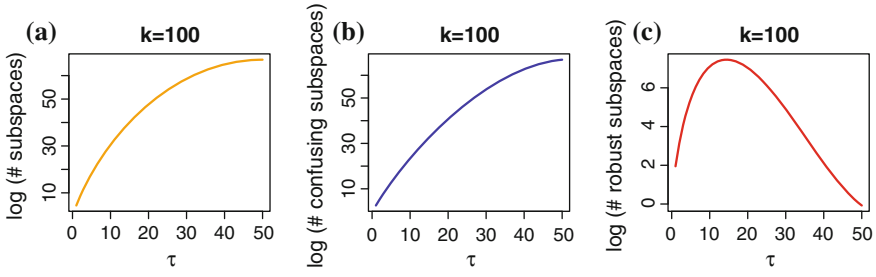


Fig. 4.3 Numbers of considered subspaces for $k = 100$

smaller. In consequence, the number of robust subspaces is maximized at around $\tau = k/2$. Conversely, smaller τ is preferable for larger noise. From this consideration, we see that sparse representation is efficient for dealing with highly contaminated data.

4.3.2 Shrinkage Estimation

The effect of regularization R can be also seen from a viewpoint of optimization process. Let us consider the gradient flow of the objective function. Figure 4.4a-c shows the contour and gradient flow of regularization terms deduced from the l_2-, l_1- and $l_{1/2}$-norms, respectively. Since our objective function is a sum of C and R, the total gradient is also a sum of individual gradients. That means the total gradient flow drastically changes depending on the property of R. The gradient flow of the l_2-regularization directs the origin as shown in Fig. 4.4a, and the coefficients diminish by adding this regularization. This effect is called shrinkage estimation which will suppress the over-fit to outliers and reduce the fluctuation of estimates. On the other hand, as shown in Fig. 4.4b, the gradient flow of the l_1-regularization directs the nearest coordinate axis; therefore, some small coefficients vanish and the rest diminish. Namely, the l_1-regularization will remove redundant coefficients and will reduce the effective degree of freedom. This phenomenon clearly appears in Fig. 4.4c for the $l_{1/2}$-regularization. Therefore, the l_p-regularization with small p conducts sparse representation naturally.

4.4 Algorithms for Dictionary Learning

As discussed in the previous section, the simple formulation of sparse representation is written as follows: For a given data matrix X, find an appropriate dictionary D and a corresponding coefficient matrix Z

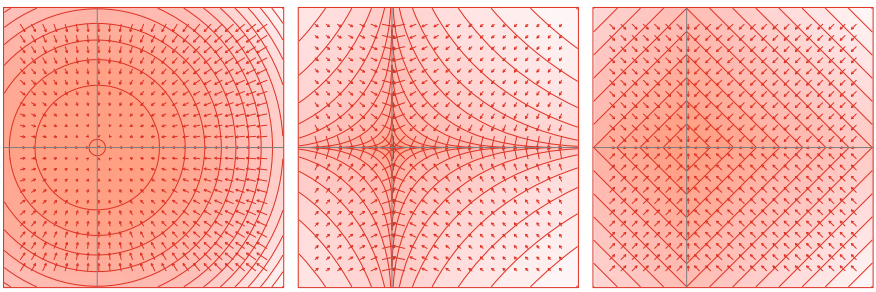


Fig. 4.4 Contours and gradient flow of l_2-, l_1- and $l_{1/2}$-regularization terms

$$\underset{D,Z}{\text{minimize}} \|X - DZ\|_2^2 + \lambda \|Z\|_s, \quad \lambda > 0,$$

where $\|\cdot\|_s$ is a certain norm imposing sparsity.

This basic problem is solved by the following two procedures repeatedly:

- estimate coefficients for a fixed dictionary,
- optimize atoms in dictionary for fixed coefficients.

In the following subsections, we discuss each procedure in detail.

4.4.1 Coefficient Estimation

The problem of estimating sparse coefficients $\mathbf{z}$ for an observation $\mathbf{x}$ is formulated as a constraint optimization problem:

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{z}\|_s \quad \text{subject to} \quad \|\mathbf{x} - D\mathbf{z}\|_2 \leq \epsilon. \tag{4.13}$$

Alternatively, we can consider the following formulation:

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{x} - D\mathbf{z}\|_2 \quad \text{subject to} \quad \|\mathbf{z}\|_s \leq \tau. \tag{4.14}$$

As well known, the solution of the above optimization problems is characterized by a saddle point of the Lagrange function

$$L(\mathbf{z}, \lambda) = \|\mathbf{x} - D\mathbf{z}\|_2^2 + \lambda \|\mathbf{z}\|_s, \quad \lambda > 0. \tag{4.15}$$

As discussed in the previous section, there are several possibilities for $\|\cdot\|_s$. Depending on the choice of sparse norms, many algorithms are proposed as follows.

Orthogonal Matching Pursuit (OMP) [29] is an iterative algorithm for finding an approximate solution of l_0-sparse representation. In each step of the algorithm, OMP selects one atom which has the largest correlation with the residual vector $\mathbf{x} - D\mathbf{z}$ in a greedy manner until desired accuracy is achieved as Eq. (4.13) or a sufficient number of atoms are collected as Eq. (4.14).

A pseudo code for solving

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{z}\|_0 \quad \text{subject to} \quad \|\mathbf{x} - D\mathbf{z}\|_2 \leq \epsilon$$

is summarized as follows.

Orthogonal Matching Pursuit

procedure OMP($D \in \mathbb{R}^{m\times k}, \mathbf{x} \in \mathbb{R}^m, \epsilon > 0$)
  $\mathbf{z} \leftarrow \mathbf{0} \in \mathbb{R}^k$ ▷ initialize coefficients
  $\mathbf{r} \leftarrow \mathbf{x} - D\mathbf{z} = \mathbf{x}$ ▷ initialize residuals
  $\mathcal{S} = \text{supp}(\mathbf{z}) \leftarrow \emptyset$ ▷ initialize support set
  while $\|\mathbf{r}\|_2 > \epsilon$ **do**
    $\delta(i) \leftarrow \min_{z_i, i\notin\mathcal{S}} \|\mathbf{r} - \mathbf{d}_i z_i\|_2^2$ ▷ calculate fitness of atoms
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\arg\min_{i\notin\mathcal{S}} \delta(i)\}$ ▷ update support set
    $\mathbf{z} \leftarrow \arg\min_{\mathbf{z},\text{supp}(\mathbf{z})\subset\mathcal{S}} \|\mathbf{x} - D\mathbf{z}\|_2^2$ ▷ update coefficients
    $\mathbf{r} \leftarrow \mathbf{x} - D\mathbf{z}$ ▷ calculate residuals
  end while
  return z
end procedure

The problem of l_1-sparse representation is equivalent to Lasso (least absolute shrinkage and selection operator) [32], and the problem is written in quadratic programming (QP) by introducing positive auxiliary variables as pointed out in [32]. If the size of the problem is quite large, QP works rather slowly; therefore, other effective methods making use of l_1-norm properties are also proposed, such as least angle regression [9], Feature Sign Search [20], and Dantzig selector [4].

A pseudo code for solving

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{x} - D\mathbf{z}\|_2^2 + \lambda\|\mathbf{z}\|_1, \quad \lambda > 0.$$

by QP with positive auxiliary variables $\mathbf{z}^+$ and $\mathbf{z}^-$ is written as follows.

Quadratic Programming for Lasso

procedure QPLASSO($D \in \mathbb{R}^{m\times k}, \mathbf{x} \in \mathbb{R}^m, \lambda > 0$)
  function $\text{L}(\mathbf{z}^+ \in \mathbb{R}_+^k, \mathbf{z}^- \in \mathbb{R}_+^k)$ ▷ define objective
    $\|\mathbf{x} - D(\mathbf{z}^+ - \mathbf{z}^-)\|_2^2 + \lambda\|\mathbf{z}^+\|_1 + \lambda\|\mathbf{z}^-\|_1$
  end function
  $\mathbf{z}^+, \mathbf{z}^- \leftarrow \arg\min_{\mathbf{z}^+,\mathbf{z}^-} \text{L}(\mathbf{z}^+, \mathbf{z}^-)$ ▷ solve by QP
  $\mathbf{z} \leftarrow \mathbf{z}^+ - \mathbf{z}^-$ ▷ calculate coefficients
  return z
end procedure

A method for solving l_p-sparse representation for general p is using an interesting relationship between the l_p-norm and l_2-norm [5, 7]. For any $p > 0$, the following simple relation holds:

$$\|\mathbf{z}\|_p^p = \|Q^{-1/2}\mathbf{z}\|_2^2, \tag{4.16}$$

where

$$Q = \mathrm{diag}(|z_1|^{2-p}, \ldots, |z_k|^{2-p}).$$

Using this relation, the optimization problem

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{z}\|_p \quad \text{subject to} \quad \mathbf{x} = D\mathbf{z}$$

is rewritten as

$$\underset{\mathbf{z}}{\text{minimize}} \|Q^{-1/2}\mathbf{z}\|_2 \quad \text{subject to} \quad \mathbf{x} = D\mathbf{z}.$$

The solution of the latter problem is given by

$$\mathbf{z} = QD^{\mathrm{T}}(DQD^{\mathrm{T}})^{-1}\mathbf{x}. \tag{4.17}$$

Since Q is a function of $\mathbf{z}$, this is not a closed form and the solution is given by the stationary point of this equation.

The following pseudo code finds the stationary point for the optimization problem of

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{z}\|_p \quad \text{subject to} \quad \|\mathbf{x} - D\mathbf{z}\|_2 \leq \epsilon.$$

Iterative Reweighted Least Squares

procedure IRLS($D \in \mathbb{R}^{m\times k}$, $\mathbf{x} \in \mathbb{R}^m$, $p > 0$, $\epsilon > 0$)
  $\mathbf{z} \leftarrow D^{\mathrm{T}}(DD^{\mathrm{T}})^{-1}\mathbf{x}$ ▷ initialize coefficients
  while $\|\mathbf{x} - D\mathbf{z}\|_2 > \epsilon$ **do**
    $Q \leftarrow \mathrm{diag}(|z_1|^{2-p}, \ldots, |z_k|^{2-p})$ ▷ calculate weights
    $\mathbf{z} \leftarrow QD^{\mathrm{T}}(DQD^{\mathrm{T}})^{-1}\mathbf{x}$ ▷ update coefficients
  end while
  return z
end procedure

4.4.2 Dictionary Optimization

For given data matrix X and coefficient matrix Z, the dictionary optimization problem is simply stated as

$$\underset{D}{\text{minimize}} \|X - DZ\|_2^2.$$

This objective is regarded as finding a good set of atoms for not specific data but all data $\mathbf{x}_i, i = 1, \ldots, n$ on average. There are many approaches adopting the gradient

methods such as original sparse coding [26, 27] and method of optimal directions [11]. Gradient methods sometimes need a lot of iterations for satisfactory convergence and are not always computationally effective. Recently, the following fast algorithms are commonly used in image analysis.

One is called K-SVD [1], which can be regarded as an extension of the k-means method for vector quantization. K-SVD updates atoms one by one in order to reduce the residuals. Corresponding coefficients are also updated during dictionary optimization. Updates of atoms are based on the singular value decomposition of residuals; therefore, the computation is fast and efficient. Let Ω be the index set of observations, i.e., $\Omega \subset \{1, \ldots, n\}$, and let X^{Ω} be a sub-matrix of X with the index set Ω, i.e., $X^{\Omega} = (\mathbf{x}_i, i \in \Omega)$. Then the pseudo code is summarized as follows.

K-SVD

procedure KSVD($D \in \mathbb{R}^{m\times k}$, $X \in \mathbb{R}^{m\times n}$, $Z \in \mathbb{R}^{k\times n}$)
  for $i = 1, \ldots, k$ **do**
    $\Omega_i \leftarrow \text{supp}(\mathbf{z}^i)$ ▷ collect data using atom d_i
    $R_i \leftarrow X^{\Omega_i} - \sum_{j=1, j\neq i}^{k} \mathbf{d}_j \mathbf{z}^{j\mathrm{T}}$ ▷ calculate residuals
    $U\Sigma V^{\mathrm{T}} \leftarrow \text{SVD}(R_i)$ ▷ calculate SVD
    $\mathbf{d}_i \leftarrow \mathbf{u}_1$ ▷ update atom d_i
    $\mathbf{z}^i \leftarrow \sigma_1 \mathbf{v}_1$ ▷ updates coefficients
  end for
  return D
end procedure

Another method is proposed by [20], which utilizes the Lagrange dual of the objective. In this method, the objective is defined as

$$\underset{D}{\text{minimize}} \|X - DZ\|_2^2 \quad \text{subject to} \quad \|\mathbf{d}_i\|_2^2 \leq \tau, i = 1, \ldots, k$$

and the Lagrange function of this problem is given by

$$L(D, \boldsymbol{\lambda}) = \text{trace}(X - DZ)^{\mathrm{T}}(X - DZ) + \sum_{i=1}^{k} \lambda_i (\|\mathbf{d}_i\|_2^2 - \tau) \tag{4.18}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_k)^{\mathrm{T}}$. By minimizing $L(D, \boldsymbol{\lambda})$ over D analytically, the Lagrange dual is obtained as

$$L(\boldsymbol{\lambda}) = \min_{D} L(D, \boldsymbol{\lambda}) = \text{trace}(X^{\mathrm{T}}X - XZ^{\mathrm{T}}(ZZ^{\mathrm{T}} + \text{diag}(\boldsymbol{\lambda}))^{-1}ZX^{\mathrm{T}} - \tau\text{diag}(\boldsymbol{\lambda})). \tag{4.19}$$

The optimal Lagrange multiplier $\boldsymbol{\lambda}$ is obtained by maximizing the Lagrange dual $L(\boldsymbol{\lambda})$, then the optimal dictionary D is analytically solved. The pseudo code is as follows.

Lagrange Dual

procedure LD($D \in \mathbb{R}^{m\times k}$, $X \in \mathbb{R}^{m\times n}$, $Z \in \mathbb{R}^{k\times n}$, $\tau > 0$)
  function L($\boldsymbol{\lambda} \in \mathbb{R}^k_+$) ▷ define Lagrange dual
    $\mathrm{trace}(X^{\mathrm{T}}X - XZ^{\mathrm{T}}(ZZ^{\mathrm{T}} + \mathrm{diag}(\boldsymbol{\lambda}))^{-1}ZX^{\mathrm{T}} - \tau\mathrm{diag}(\boldsymbol{\lambda}))$
  end function
  $\boldsymbol{\lambda} \leftarrow \arg\max_{\boldsymbol{\lambda}} \mathrm{L}(\boldsymbol{\lambda})$ ▷ solve by, e.g., Newton method
  $D \leftarrow XZ^{\mathrm{T}}(ZZ^{\mathrm{T}} + \mathrm{diag}(\boldsymbol{\lambda}))^{-1}$ ▷ update dictionary
  return D
end procedure

4.5 Applications to Image Processing

In signal processing, there are many applications of matrix factorization technique such as

- noise reduction and restoration [10, 24],
- face recognition [34],
- super-resolution [18, 35],
- image decomposition and separation [12],
- biological signal processing [6],
- recommendation system [19],
- text categorization [2].

We present noise reduction and restoration, face recognition, and super-resolution as typical applications of sparse matrix factorization.

Noise Reduction and Restoration

Noise reduction or denoising is a basic and popular application of sparse representation in image processing [10, 24].

First, we prepare an appropriate dictionary D learned from a plenty of real-world images which belong to the same category of the target image, such as natural scenes and human faces. Then, the target image $\mathbf{x}$ is approximated by

$$\underset{\mathbf{z}}{\text{minimize}}\ \|\mathbf{x} - D\mathbf{z}\|_2 \quad \text{subject to} \quad \|\mathbf{z}\|_s \leq \tau,$$

where τ is defined in the dictionary learning stage. If the noise level ϵ is known beforehand, we can apply

$$\underset{\mathbf{z}}{\text{minimize}} \|\mathbf{z}\|_s \quad \text{subject to} \quad \|\mathbf{x} - D\mathbf{z}\|_2 \leq \epsilon.$$

In any cases, because of the nature of coefficient sparsity, it is expected that $D\mathbf{z}$ does not include unusual components, namely noisy components, which cannot be represented by usual atoms learned from real-world images.

In the case that the observed image lacks some of the pixels, we can apply sparse representation with a small modification. Let $R_{\mathbf{x}}$ be an operator that removes missing elements of $\mathbf{x}$. Then, the above noise reduction procedure is rewritten as

$$\underset{\mathbf{z}}{\text{minimize}} \|R_{\mathbf{x}}(\mathbf{x} - D\mathbf{z})\|_2 \quad \text{subject to} \quad \|\mathbf{z}\|_s \leq \tau.$$

In this formulation, the coefficient vector $\mathbf{z}$ is estimated only from the observed part, and the missing part is complemented by atoms with the estimated coefficients.

Face Recognition

For discriminative task such as face recognition, design of the dictionary is quite important. There are many approaches, but a simple one is using sample images of target users as atoms [34]. Suppose U users have registered their face images. Let $\mathbf{d}_i^u$ be the ith face image of user u, and k_u be the number of u's face images. Then, we construct a dictionary as

$$D = (\mathbf{d}_1^1, \ldots, \mathbf{d}_{k_1}^1, \ldots, \mathbf{d}_1^u, \ldots, \mathbf{d}_{k_u}^u, \ldots, \mathbf{d}_1^U, \ldots, \mathbf{d}_{k_U}^U).$$

If the target image $\mathbf{x}$ is a registered user's face image, then its sparse representation $\mathbf{z}$ for D is expected to have nonzero elements at the index of the corresponding user. Therefore, the estimated coefficients can be used as features for various kinds of classifiers.

Super-Resolution

Super-resolution is a technique for recovering an original high-resolution image $\mathbf{x}^H$ from an observation $\mathbf{x}^L$ degraded to low-resolution. The key idea is that if we prepare a proper pair of high-resolution dictionary D^H and low-resolution dictionary D^L, the original image $\mathbf{x}^H$ and degraded image $\mathbf{x}^L$ can be represented by the same coefficients $\mathbf{z}$ as

$$\mathbf{x}^H = D^H\mathbf{z},$$
$$\mathbf{x}^L = D^L\mathbf{z}.$$

In the case of single-frame super-resolution, where a high-resolution image is recovered from only one single low-resolution image, both of the dictionaries are simultaneously trained by a set of high-resolution images and low-resolution images sustained by the same degradation process [35]. In the case of multi-frame super-resolution, where differently degraded several low-resolution images are given for recovering a high-resolution image, we need an additional process of image patch

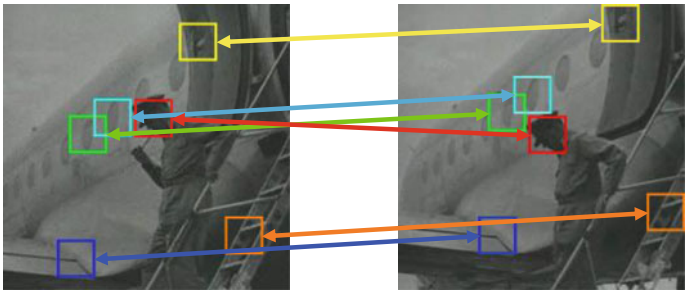
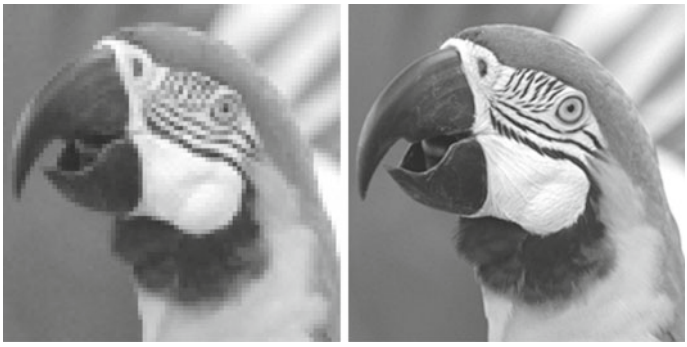


Fig. 4.5 Concept of image patch alignment



degraded image recovered image

Fig. 4.6 Super-resolution of degraded image

alignment as shown in Fig. 4.5. By stacking well-aligned low-resolution images $\mathbf{x}_i^L$, $i = 1, \ldots, l$ and corresponding dictionaries D_i^L, $i = 1, \ldots, l$ as

$$\tilde{\mathbf{x}}^L = \begin{pmatrix} \mathbf{x}_1^L \\ \vdots \\ \mathbf{x}_l^L \end{pmatrix}, \quad \tilde{D}^L = \begin{pmatrix} D_1^L \\ \vdots \\ D_l^L \end{pmatrix},$$

we can use the above single-frame strategy [18]. Figure 4.6 shows an example of super-resolution with sparse representation.

Acknowledgments The author greatly appreciates Dr. Hideitsu Hino of University of Tsukuba and Mr. Toshiyuki Kato of Waseda University for their helpful comments and figures in Sects. 4.2 and 4.5.

References

1. Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, *54*(11), 4311–4322.
2. Berry, M. W., & Browne, M. (2005). Email surveillance using non-negative matrix factorization. *Computational and Mathematical Organization Theory*, *11*(3), 249–264.
3. Candès, E. J., & Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, *51*(12), 4203–4215.
4. Candès, E. J., & Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, *35*(6), 2313–2351.
5. Chartrand, R., & Yin, W. (2008). Iteratively reweighted algorithms for compressive sensing. In: *ICASSP* (pp. 3869–3872).
6. Cichocki, A., Zdunek, R., Phan, A.H., & Amari, S. (2009). *Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation*. Wiley.
7. Daubechies, I., DeVore, R., Fornasier, M., & Güntürk, C. S. (2010). Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, *63*(1), 1–38.
8. Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, *52*(4), 1289–1306.
9. Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics,32*(2), 407–499.
10. Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, *15*(12), 3736–3745.
11. Engan, K., Aase, S. O., & Husϕy, J. H. (1999). Method of optimal directions for frame design. In: *ICASSP* (pp. 2443–2446).
12. Fadili, M. J., Starck, J. L., Bobin, J., & Moudden, Y. (2010). Image decomposition and separation using sparse representations: An overview. *Proceedings of the IEEE*, *98*(6), 983–994.
13. Fujimoto, Y., & Murata, N. (2012). Nonnegative matrix factorization via generalized product rule and its application for classification. *LVA/ICA, LNCS*, *7191*, 263–271.
14. Harman, H. H. (1976). *Modern Factor Analysis* (3rd ed.). University of Chicago Press.
15. Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research,5*, 1457–1469.
16. Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent Component Analysis*. Wiley.
17. Jolliffe, I. T. (2002). *Principal Component Analysis* (2nd ed.). Springer Series in Statistics. New York: Springer.
18. Kato, T., Hino, H., & Murata, N. (2015). Multi-frame image super resolution based on sparse coding. *Neural Networks*, *66*, 64–78.
19. Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.
20. Lee, H., Battle, A., Raina, R., & Ng, A. Y. (2006). Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, *19*, 801–808.
21. Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.
22. Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, *13*, 556–562.
23. Li, S. Z. (2009). *Markov Random Field Modeling in Image Analysis: Advances in Pattern Recognition*. London: Springer.
24. Mairal, J., Elad, M., & Sapiro, G. (2008). Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, *17*(1), 53–69.
25. Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, *24*(2), 227–234.

26. Olshausen, B.A., & Field, D.J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1?. *Vision Research*, *37*(23), 3311–3325.
27. Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*, 607–609.
28. Paatero, P., & Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, *5*(2), 111–126.
29. Pati, Y., Rezaiifar, R., & Krishnaprasad, P. (1993). Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Signals, Systems and Computers*, *1*, 40–44.
30. Phillips, P. J., Wechsler, H., Huang, J., & Rauss, P. J. (1998). The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, *16*, 295–306.
31. Shashua, A., & Hazan, T. (2005). Non-negative tensor factorization with applications to statistics and computer vision. In *ICML*, (pp. 792–799).
32. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, *58*(1), 267–288.
33. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.
34. Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.
35. Yang, J., Wright, J., Huang, T. S., & Ma, Y. (2010). Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, *19*(11), 2861–2873.

Chapter 5
Array Normal Model and Incomplete Array Variate Observations

Deniz Akdemir

Abstract Missing data present an important challenge when dealing with high-dimensional data arranged in the form of an array. The main purpose of this article is to introduce methods for estimation of the parameters of array variate normal probability model from partially observed multiway data. The methods developed here are useful for missing data imputation, estimation of mean, and covariance parameters for multiway data. A review of array variate distributions is included. A multiway semi-parametric mixed-effects model that allows separation of multiway mean and covariance effects is also defined, and an efficient algorithm for estimation based on the spectral decompositions of the covariance parameters is recommended. We demonstrate our methods with simulations and real-life data involving the estimation of genotype and environment interaction effects on possibly correlated traits.

Keywords Array normal distribution · Missing data · Imputation · Kronecker covariance structure · Mixed models · Kernel-based learning

5.1 Introduction

A vector is a one-way array, a matrix is a two-way array, by stacking matrices, we obtain three way arrays, etc. Array variate random variables up to two dimensions have been studied intensively in [13] and by many others. For array observations of 3, 4 or in general i dimensions, probability models with Kronecker delta covariance structure have been proposed very recently in [1, 24, 33]. The estimation and inference for the parameters of the array normal distribution with Kronecker delta covariance structure, based on a random sample of fully observed arrays $\{\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_N\}$, can be accomplished by maximum-likelihood estimation [1, 24, 33, 34] or by Bayesian estimation [18].

Array variate random variables are mainly useful for multiply labeled random variables that can naturally be arranged in array form. Some examples include response

D. Akdemir (✉)
Department of Plant Breeding and Genetics, Cornell University, Ithaca, NY, USA
e-mail: akdemir.work@gmail.com

T. Sakata (ed.), *Applied Matrix and Tensor Variate Data Analysis*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55387-8_5

from multifactor experiments, two- to three-dimensional image–video data, spatial–temporal data, and repeated measures data. For example, in a typical plant breeding experiment, a four-way array data can be obtained by considering the varieties × attributes × locations × years. It is true that any array data can also be represented uniquely in vector form, and a general covariance structure can be assumed for this vector representation. However, the models with Kronecker covariance structure are relatively more parsimonious.

The array variate data models and the estimation techniques we have mentioned above assume that we have a random sample of fully observed arrays. However, in practice, most array data come with many missing cells. The main purpose of this article is to develop methods for estimation and inference for a class of array random variables when we only have partially observed arrays in the random sample.

Another novelty in this article involves the definition and development of a multiway mixed-effects model. This model is useful for analyzing multiway response variables that depend on separable effects, and through it, we can incorporate the known covariance structures along some dimensions of the response and we can estimate the unknown mean and covariance components.

The array variate mixed models can be used to fit Gaussian process regression models with multiway data. Using the explanatory information that describes levels related to the dimension of an array, we can calculate a kernel matrix for that dimension. The shrinkage parameters related to a kernel along a dimension can be estimated using likelihood-based methods. Similarly, the covariance for the dimensions with no explanatory information can also be estimated. We illustrate this with two examples where we calculate and use kernel matrices based on genetic information in the form of genomewide markers.

The remaining of the article is organized as follows: In Sects. 5.2 and 5.3, we introduce the normal model for array variables. In Sect. 5.4, we introduce the updating equations for parameter estimation and missing data imputation. In Sect. 5.5, the basic algorithm is introduced. In Section 5.6, we define a semi-parametric array variate mixed model with Kronecker covariance structure, and an efficient algorithm for the estimation of variance components is described. In Sect. 5.7, we study the principal component analysis for the array case. Section 5.8 includes a brief discussion of classification. Examples illustrating the use of these methods are provided in Sect. 5.9, followed by our conclusions in Sect. 5.10.

5.2 Arrays and Array Variate Random Variables

In this paper, we will only study arrays with real elements. We will write $\widetilde{X}$ to say that $\widetilde{X}$ is an array. When it is necessary, we can write the dimensions of the array as subindices, e.g., if $\widetilde{X}$ is a $m_1 \times m_2 \times m_3 \times m_4$ dimensional array in $R^{m_1 \times m_2 \times \ldots \times m_i}$, then we can write $\widetilde{X}_{m_1 \times m_2 \times m_3 \times m_4}$. To refer to an element of an array $\widetilde{X}_{m_1 \times m_2 \times m_3 \times m_4}$, we write the position of the element as a subindex to the array name in parenthesis, $(\widetilde{X})_{r_1 r_2 r_3 r_4}$.

Arrays can be constant arrays, i.e., if $(\widetilde{X})_{r_1 r_2 \ldots r_i} \in \mathbf{R}$ are constants for all r_j, $j = 1, 2, \ldots, m_j$ and $j = 1, 2, \ldots, i$, then the array $\widetilde{X}$ is a constant array.

Array variate random variables are arrays with all elements $(\widetilde{X})_{r_1 r_2 \ldots r_i} \in \mathbf{R}$ random variables. If the sample space for the random outcome s is $\mathbb{S}$, $(\widetilde{X})_{r_1 r_2 \ldots r_i} = (\widetilde{X}(s))_{r_1 r_2 \ldots r_i}$ where each of $(\widetilde{X}(s))_{r_1 r_2 \ldots r_i}$ is a real-valued function from $\mathbb{S}$ to $\mathbb{R}$.

If $\widetilde{X}$ is an array variate random variable, its density (if it exists) is a scalar function $f_{\widetilde{X}}(\widetilde{X})$ such that:

- $f_{\widetilde{X}}(\widetilde{X}) \geq 0$;
- $\int_{\widetilde{X}} f_{\widetilde{X}}(\widetilde{X}) d\widetilde{X} = 1$;
- $P(\widetilde{X} \in A) = \int_A f_{\widetilde{X}}(\widetilde{X}) d\widetilde{X}$, where A is a subset of the space of realizations for $\widetilde{X}$.

A scalar function $f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y})$ defines a joint (biarray variate) probability density function if

- $f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y}) \geq 0$;
- $\int_{\widetilde{Y}} \int_{\widetilde{X}} f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y}) d\widetilde{X} d\widetilde{Y} = 1$;
- $P((\widetilde{X}, \widetilde{Y}) \in A) = \int \int_A f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y}) d\widetilde{X} d\widetilde{Y}$, where A is a subset of the space of realizations for $(\widetilde{X}, \widetilde{Y})$.

The marginal probability density function of $\widetilde{X}$ is defined by

$$f_{\widetilde{X}}(\widetilde{X}) = \int_{\widetilde{Y}} f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y}) d\widetilde{Y},$$

and the conditional probability density function of $\widetilde{X}$ given $\widetilde{Y}$ is defined by

$$f_{\widetilde{X}|\widetilde{Y}}(\widetilde{X}|\widetilde{Y}) = \frac{f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y})}{f_{\widetilde{Y}}(\widetilde{Y})},$$

where $f_{\widetilde{Y}}(\widetilde{Y}) > 0$.

Two random arrays $\widetilde{X}$ and $\widetilde{Y}$ are independent if and only if

$$f_{\widetilde{X},\widetilde{Y}}(\widetilde{X}, \widetilde{Y}) = f_{\widetilde{X}}(\widetilde{X}) f_{\widetilde{Y}}(\widetilde{Y}).$$

It is well known that a matrix equation

$$AXB' = C$$

can be rewritten in its monolinear form as

$$A \otimes^i Bvec(X) = vec(C). \tag{5.1}$$

Furthermore, the matrix equality

$$A \otimes^i BXC' = E$$

obtained by stacking equations of the form (5.1) can be written in its monolinear form as

$$(A \otimes^i B \otimes^i C)vec(X) = vec(E).$$

This process of stacking equations could be continued, and R-matrix multiplication operation introduced by Rauhala [26] provides a compact way of representing these equations in array form:

Definition 1 *R-matrix multiplication* is defined elementwise:

$$((A_1)^1(A_2)^2 \dots (A_i)^i \widetilde{X}_{m_1 \times m_2 \times \dots \times m_i})_{q_1 q_2 \dots q_i}$$

$$= \sum_{r_1=1}^{m_1} (A_1)_{q_1 r_1} \sum_{r_2=1}^{m_2} (A_2)_{q_2 r_2} \sum_{r_3=1}^{m_3} (A_3)_{q_3 r_3} \dots \sum_{r_i=1}^{m_i} (A_i)_{q_i r_i} (\widetilde{X})_{r_1 r_2 \dots r_i}.$$

R-matrix multiplication generalizes the matrix multiplication (array multiplication in two dimensions) to the case of k-dimensional arrays. The following useful properties of the R-matrix multiplication are reviewed by Blaha [6]:

- $(A)^1 B = AB$.
- $(A_1)^1(A_2)^2 C = A_1 C A_2'$.
- $\widetilde{Y} = (I)^1(I)^2 \dots (I)^i \widetilde{Y}$.
- $((A_1)^1(A_2)^2 \dots (A_i)^i)((B_1)^1(B_2)^2 \dots (B_i)^i)\widetilde{Y} = (A_1 B_1)^1(A_2 B_2)^2 \dots (A_i B_i)^i \widetilde{Y}$.

The operator $rvec$ describes the relationship between $\widetilde{X}_{m_1 \times m_2 \times \dots \times m_i}$ and its monolinear form $\boldsymbol{x}_{m_1 m_2 \dots m_i \times 1}$.

Definition 2 $rvec(\widetilde{X}_{m_1 \times m_2 \times \dots m_i}) = \boldsymbol{x}_{m_1 m_2 \dots m_j \times 1}$ where $\boldsymbol{x}$ is the column vector obtained by stacking the elements of the array $\widetilde{X}$ in the order of its dimensions; i.e., $(\widetilde{X})_{j_1 j_2 \dots j_i} = (\boldsymbol{x})_j$ where $j = (j_i - 1)n_{i-1}n_{i-2} \dots n_1 + (j_i - 2)n_{i-2}n_{i-3} \dots n_1 + \dots + (j_2 - 1)n_1 + j_1$.

Let $\widetilde{L}_{m_1 \times m_2 \times \dots \times m_i} = (A_1)^1(A_2)^2 \dots (A_i)^i \widetilde{X}$ where $(A_j)^j$ is an $m_j \times n_j$ matrix for $j = 1, 2, \dots, i$ and $\widetilde{X}$ is an $n_1 \times n_2 \times \dots \times n_i$ array. Write $\mathbf{l} = rvec(\widetilde{L})$ and $\boldsymbol{x} = rvec(\widetilde{X})$. Then, $\mathbf{l} = A_1 \otimes^i A_2 \otimes^i \dots \otimes^i A_i \boldsymbol{x}$. Therefore, there is an equivalent expression of the array equation in monolinear form.

Definition 3 The square norm of $\widetilde{X}_{m_1 \times m_2 \times \dots \times m_i}$ is defined as

$$\|\widetilde{X}\|^2 = \sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \dots \sum_{j_i=1}^{m_i} ((\widetilde{X})_{j_1 j_2 \dots j_i})^2.$$

Definition 4 The distance of $\widetilde{X}_{1m_1\times m_2\times\ldots\times m_i}$ from $\widetilde{X}_{2m_1\times m_2\times\ldots\times m_i}$ is defined as

$$\sqrt{\|\widetilde{X}_1-\widetilde{X}_2\|^2}.$$

Example 5.2.1 Let $\widetilde{Y}=(A_1)^1(A_2)^2\ldots(A_i)^i\widetilde{X}+\widetilde{E}$. Then $\|\widetilde{E}\|^2$ is minimized for $\widehat{\widetilde{X}}=(A_1^-)^1\ (A_2^-)^2\ldots(A_i^-)^i\ \widetilde{Y}$.

Theorem 5.2.1 *Let* $(A_1)^1$, $(A_2)^2$, $\ldots$, $(A_i)^i$ *be* $m_1, m_2, \ldots, m_i$ *dimensional positive definite matrices. The Jacobian* $J(\widetilde{X}\to\widetilde{Z})$ *of the transformation* $\widetilde{X}=(A_1)^1(A_2)^2\ldots(A_i)^i\widetilde{Z}+\widetilde{M}$ *is*

$$(|A_1|^{\prod_{j\neq1}m_j}|A_2|^{\prod_{j\neq2}m_j}\ldots|A_i|^{\prod_{j\neq i}m_j})^{-1}.$$

Proof The result is proven using the equivalence of monolinear form obtained through the $rvec(\widetilde{X})$ and array $\widetilde{X}$. Let $\widetilde{L}_{m_1\times m_2\times\ldots m_j}=(A_1)^1(A_2)^2\ldots(A_i)^i\widetilde{Z}$ where $(A_j)^j$ is an $m_j\times n_j$ matrix for $j=1,2,\ldots,i$ and $\widetilde{X}$ is an $n_1\times n_2\times\ldots\times n_i$ array. Write $\mathbf{l}=rvec(\widetilde{L})$ and $\boldsymbol{z}=rvec(\widetilde{Z})$. Then, $\mathbf{l}=A_1\otimes^i A_2\otimes^i\ldots\otimes^i A_i\boldsymbol{z}$. The result follows from noting that $J(\mathbf{l}\to\boldsymbol{z})=|A_1\otimes^i A_2\otimes^i\ldots\otimes^i A_i|^{-1}$ and using induction with the rule $|A\otimes^i B|=|A|^m|B|^n$ for $n\times n$ matrix A and $m\times m$ matrix B to show that $|A_1\otimes^i A_2\otimes^i\ldots\otimes^i A_i|^{-1}=(|A_1|^{\prod_{j\neq1}m_j}|A_2|^{\prod_{j\neq2}m_j}\ldots|A_i|^{\prod_{j\neq i}m_j})^{-1}$.

Corollary 5.2.1 *Let* $\widetilde{Z}\sim f_{\widetilde{Z}}(\widetilde{Z})$. *Define* $\widetilde{X}=(A_1)^1(A_2)^2\ldots(A_i)^i\widetilde{Z}+\widetilde{M}$ *where* $(A_1)^1$, $(A_2)^2$, $\ldots$, $(A_i)^i$ *are* $m_1, m_2, \ldots, m_i$ *dimensional positive definite matrices. The pdf of* $\widetilde{X}$ *is given by*

$$f_{\widetilde{X}}(\widetilde{X};(A_1)^1,(A_2)^2,\ldots,(A_i)^i,\widetilde{M})=\frac{f(A_1^{-1})^1(A_2^{-1})^2\ldots(A_i^{-1})^i(\widetilde{X}-\widetilde{M}))}{|A_1|^{\prod_{j\neq1}m_j}|A_2|^{\prod_{j\neq2}m_j}\ldots|A_i|^{\prod_{j\neq i}m_j}}.$$

5.3 Array Normal Random Variable

The family of normal densities with Kronecker delta covariance structure is given by

$$\phi(\widetilde{X};\widetilde{\mathcal{M}},\mathcal{A}_1,\mathcal{A}_2,\ldots\mathcal{A}_i)=\frac{\exp(-\frac{1}{2}\|(\mathcal{A}_1^{-1})^1(\mathcal{A}_2^{-1})^2\ldots(\mathcal{A}_i^{-1})^i(\widetilde{X}-\widetilde{\mathcal{M}})\|^2)}{(2\pi)^{(\prod_j m_j)/2}|\mathcal{A}_1|^{\prod_{j\neq1}m_j}|\mathcal{A}_2|^{\prod_{j\neq2}m_j}\ldots|\mathcal{A}_i|^{\prod_{j\neq i}m_j}}\tag{5.2}$$

where $\mathcal{A}_1,\mathcal{A}_2,\ldots,\mathcal{A}_i$ are non-singular matrices of orders $m_1, m_2, \ldots, m_i$.

The construction of this density follows mainly from the relationship of the arrays to their monolinear forms described by Definition 2. For the scalar case, the density of the standard normal variable $z\in\mathbf{R}^1$ is given as

$$\phi_1(z)=\frac{1}{(2\pi)^{\frac{1}{2}}}exp(-\frac{1}{2}z^2).$$

For the m_1 dimensional standard normal vector $z \in \mathbf{R}^{m_1}$, the density is given by

$$\phi_{m_1}(z) = \frac{1}{(2\pi)^{\frac{m_1}{2}}} exp(-\frac{1}{2}z'z).$$

Finally, the $m_1 \times m_2$ standard matrix variate variable $Z \in \mathbf{R}^{m_1 \times m_2}$ has the density

$$\phi_{m_1 \times m_2}(Z) = \frac{1}{(2\pi)^{\frac{m_1 m_2}{2}}} exp(-\frac{1}{2}trace(Z'Z)).$$

With the above definition, we have generalized the notion of normal random variable to the array variate case.

Definition 5 We write

$$\widetilde{X} \sim N_{m_1 \times m_2 \times \ldots \times m_i}(\widetilde{M}, \Lambda_{m_1 m_2 \ldots m_i})$$

if $rvec(\widetilde{X}) \sim N_{m_1 m_2 \ldots m_i}(rvec(\widetilde{M}), \Lambda_{m_1 m_2 \ldots m_i})$. Here, $\widetilde{M}$ is the expected value of $\widetilde{X}$, and $\Lambda_{m_1 m_2 \ldots m_i}$ is the covariance matrix of the $m_1 m_2 \ldots m_i$-variate random variable $rvec(\widetilde{X})$.

The family of normal densities with Kronecker delta covariance structure is obtained by considering the densities obtained by the location-scale transformations of the standard normal variables.

Other array variate densities in the elliptical family are easily constructed using Corollary 5.2.1. For example, the following definition provides a generalization of the Student's t distribution to the array variate case.

Let $A_1, A_2, \ldots, A_i$ be non-singular matrices with orders $m_1, m_2, \ldots, m_i$ and $\widetilde{M}$ be a $m_1 \times m_2 \times \ldots \times m_i$ constant array. Then, the pdf of an $m_1 \times m_2 \times \ldots \times m_i$ array variate t random variable, $\widetilde{T}$, with degrees of freedom k given by

$$f(\widetilde{T}; \widetilde{M}, A_1, A_2, \ldots A_i) = c \frac{(1 + \|(A_1^{-1})^1 (A_2^{-1})^2 \ldots (A_i^{-1})^i (\widetilde{T} - \widetilde{M})\|^2)^{-(k+m_1 m_2 \ldots m_i)/2}}{|A_1|^{\prod_{j \neq 1} m_j} |A_2|^{\prod_{j \neq 2} m_j} \ldots |A_i|^{\prod_{j \neq i} m_j}} \quad (5.3)$$

where $c = \frac{(k\pi)^{m_1 m_2 \ldots m_i/2} \Gamma((k+m_1 m_2 \ldots m_i)/2)}{\Gamma(k/2)}$.

Distributional properties of an array normal variable with density in the form of Eq. 5.2 can be obtained by using the equivalent monolinear representation. The moments, the marginal and conditional distributions, and independence of variates should be studied considering the equivalent monolinear form of the array variable and the well-known properties of the multivariate normal random variable.

An important operation with an array is the matricization (also known as unfolding or flattening) operation, and it is the process of arranging the elements of an array in a matrix. Matricization of an array of dimensions $m_1, \times m_2, \ldots, m_i$ along its kth dimension is obtained by stacking the m_k dimensional column vectors along the kth in the order of the levels of the other dimensions and results in a $m_k \times \prod_{j \neq k} m_j$ matrix.

The operator $rvec$ describes the relationship between $\widetilde{X}_{m_1\times m_2\times\ldots\times m_i}$ and its monolinear form $\boldsymbol{x}_{m_1m_2\ldots m_i\times 1}$. $rvec(\widetilde{X}_{m_1\times m_2\times\ldots\times m_i}) = \boldsymbol{x}_{m_1m_2\ldots m_i\times 1}$ where $\boldsymbol{x}$ is the column vector obtained by stacking the elements of the array $\widetilde{X}$ in the order of its dimensions; i.e., $(\widetilde{X})_{j_1j_2\ldots j_i} = (\boldsymbol{x})_j$ where $j = (j_i - 1)m_{i-1}m_{i-2}\ldots m_1 + (j_i - 2)m_{i-2}m_{i-3}\ldots m_1 + \ldots + (j_2 - 1)m_1 + j_1$.

The following are very useful properties of the array normal variable with Kronecker delta covariance structure [1]. Property 5.2 describes the relationship of the multilinear array normal distribution to the multivariate normal distribution.

Property 5.1 *If* $\widetilde{X} \sim \phi(\widetilde{X};\ \widetilde{\mathcal{M}},\ \mathcal{A}_1,\ \mathcal{A}_2,\ \ldots\ \mathcal{A}_i)$, *then* $rvec(\widetilde{X}) \sim \phi(rvec(\widetilde{X});\ rvec(\widetilde{\mathcal{M}}),\ \mathcal{A}_i \otimes \ldots \otimes \mathcal{A}_2 \otimes \mathcal{A}_1)$.

Property 5.2 *If* $\widetilde{X} \sim \phi(\widetilde{X}; \widetilde{\mathcal{M}}, \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i)$, *then* $E(rvec(\widetilde{X})) = rvec(\widetilde{\mathcal{M}})$ *and* $cov(rvec(\widetilde{X})) = (\mathcal{A}_i \otimes \ldots \otimes \mathcal{A}_2 \otimes \mathcal{A}_1)(\mathcal{A}_i \otimes \ldots \otimes \mathcal{A}_2 \otimes \mathcal{A}_1)'$.

Inference about the parameters of the model in (5.2) for the matrix variate case has been considered in the statistical literature [22, 28, 29, 34], etc. The flip-flop algorithm [34] is proven to attain maximum-likelihood estimators of the parameters of two-dimensional array variate normal distribution. In [1, 18, 24], the flip-flop algorithm was extended to general array variate case.

The following is similar to the flip-flop algorithm. First, assume $\{\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_N\}$ is a random sample from a $N(\widetilde{M}, A_1, A_2, \ldots A_i)$ distribution with $j-1$ of the last diagonal elements of matrices A_jA_j' equal to 1 for $j = 1, 2, \ldots, i$. Further, we assume that all A_j's are square positive definite matrices of rank at least j. Finally, assume that we have $N\prod_{j=1}^{i} m_j > m_r^2$ for all $r = 1, 2, \ldots, i$.

Algorithm for estimation is as follows:

1. Estimate $\widetilde{M}$ by $\widehat{\widetilde{M}} = \frac{1}{N}\sum_{l=1}^{N}\widetilde{X}_l$.
2. Start with initial estimates of $A_2, A_3, \ldots, A_i$.
3. On the basis of the estimates of $A_2, A_3, \ldots, A_i$, calculate an estimate of A_1 by first scaling the array observations using

$$\widetilde{Z}_l = (I)^1(A_2^{-1})^2(A_3^{-1})^3\ldots(A_i^{-1})^i(\widetilde{X}_l - \widehat{\widetilde{M}}),$$

and setting

$$\boldsymbol{\Sigma}_1 = \frac{1}{N\prod_{j\neq k} m_j}\sum_{q=1}^{N\prod_{j\neq k} m_j} Z_{(k)q}Z'_{(k)q}$$

and then calculating the square root of covariance along the 1st dimension of the arrays $\widetilde{Z}_l$, $l = 1, 2, \ldots, N$.

4. On the basis of the most recent estimates of the model parameters, estimate A_j $j = 2, \ldots, i$. by first scaling the array observations using

$$\widetilde{Z}_l = (A_1^{-1})^1(A_2^{-1})^2\ldots(A_{j-1}^{-1})^{j-1}I(A_{j+1}^{-1})^{j+1}\ldots(A_i^{-1})^i(\widetilde{X}_l - \widehat{\widetilde{M}}),$$

and then calculating the square root of covariance along the jth dimension of the arrays $\widetilde{Z}_l$'s for $j = 2, \ldots, i$. Scale the estimate of $\boldsymbol{\Sigma}_j$ so that the first diagonal is equal to 1.

5. Repeat steps 3 and 4 until convergence is attained.

In the remaining of this paper, we will assume that the matrices $\mathcal{A}_i$ are unique square roots (e.g., eigenvalue or Cholesky decompositions) of the positive definite matrices $\boldsymbol{\Sigma}_i$ for $i = 1, 2, \ldots, i$ and we will put $\Lambda = \boldsymbol{\Sigma}_i \otimes \ldots \otimes \boldsymbol{\Sigma}_2 \otimes \boldsymbol{\Sigma}_1 = (\mathcal{A}_i \otimes \ldots \otimes \mathcal{A}_2 \otimes \mathcal{A}_1)(\mathcal{A}_i \otimes \ldots \otimes \mathcal{A}_2 \otimes \mathcal{A}_1)'$ for the overall covariance matrix.

We also use the following notation:

- $$(A)^k\widetilde{X} \equiv (I)^1(I)^2 \ldots (I)^{k-1}(A)^k(I)^{k+1} \ldots (I)^i\widetilde{X}.$$
- For vectors $\boldsymbol{a}_k$, $k = 1, 2, \ldots, i$,
$$(\boldsymbol{a}_1)^1(\boldsymbol{a}_2)^2 \ldots (\boldsymbol{a}_i)^i 1 \equiv (\boldsymbol{a}_1)^1(\boldsymbol{a}_2)^2 \ldots (\boldsymbol{a}_i)^i\widetilde{1}_{1\times 1\times \ldots \times 1}.$$
- Matricization of $\widetilde{X}$ along the kth dimension: $X_{(k)}$
- For ease of notation, when the dimensions are evident from the context, we have used $\mathbf{0}$ to stand for the zero matrix with appropriate dimensions.
- A vector of ones: $\mathbf{1}$.

5.4 Dealing with Incomplete Arrays

Using linear predictors for the purpose of imputing missing values in multivariate normal data dates back at least as far as [3]. The EM algorithm [8] is usually utilized for multivariate normal distribution with missing data. The EM method goes back to [5, 25]. Hartley and Hocking [15, 35] developed the Fisher scoring algorithm for incomplete multivariate normal data. The notation and the algorithms described in this section were adopted from [19].

Let $\boldsymbol{x}$ be a k-dimensional observation vector which is partitioned as

$$\begin{bmatrix} R \\ M \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_m \end{bmatrix}$$

where $\boldsymbol{x}_r$ and $\boldsymbol{x}_m$ represent the vector of observed values and the missing observations correspondingly. Here,

$$\begin{bmatrix} R \\ M \end{bmatrix}$$

is an orthogonal permutation matrix of zeros and ones and

$$\boldsymbol{x} = \begin{bmatrix} R \\ M \end{bmatrix}' \begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_m \end{bmatrix}.$$

The mean vector and the covariance matrix of $\begin{bmatrix} \boldsymbol{x}_r \\ \boldsymbol{x}_m \end{bmatrix}$ are given by

$$\begin{bmatrix} R \\ M \end{bmatrix} E(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\mu}_r \\ \boldsymbol{\mu}_m \end{bmatrix}$$

and

$$\begin{bmatrix} R \\ M \end{bmatrix} cov(\boldsymbol{x}) \begin{bmatrix} R \\ M \end{bmatrix}' = \begin{bmatrix} \boldsymbol{\Sigma}_{rr} & \boldsymbol{\Sigma}_{rm} \\ \boldsymbol{\Sigma}_{mr} & \boldsymbol{\Sigma}_{mm} \end{bmatrix}$$

correspondingly.

Let $\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_N$ be a random sample of array observations from the distribution with density $\phi(\widetilde{X}; \widetilde{\mathcal{M}}, \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i)$. Let the current values of the parameters be $\widetilde{\mathcal{M}}^t, \mathcal{A}_1^t, \mathcal{A}_2^t, \ldots \mathcal{A}_i^t$.

The mean of the conditional distribution of $rvec(\widetilde{X}_l)$ given the estimates of parameters at time t ($t = 0, 1, 2, \ldots$) can be obtained using

$$rvec(\widehat{\widetilde{X}}_l^{t+1}) = rvec\widetilde{\mathcal{M}}^t + \Lambda^t R_l'(R_l \Lambda^t R_l')^{-1}(R_l \boldsymbol{x}_l - R_l rvec(\widetilde{\mathcal{M}}^t)) \tag{5.4}$$

where $\boldsymbol{x}_l = rvec(\widetilde{X}_l)$ and R_l is the permutation matrix such that $\boldsymbol{x}_{rl} = R_l \boldsymbol{x}_l$. The updating equation of the parameter $\widetilde{\mathcal{M}}$ is given by

$$rvec(\widetilde{\mathcal{M}}^{t+1}) = \frac{1}{N} \sum_{l=1}^{N} rvec(\widehat{\widetilde{X}}_l^{t+1}). \tag{5.5}$$

To update the covariance matrix along the kth dimension, calculate

$$\widetilde{Z} = (\mathcal{A}_1^{-1})^1 (\mathcal{A}_2^{-1})^2 \ldots (\mathcal{A}_{k-1}^{-1})^{k-1} (I_{m_k})^k (\mathcal{A}_{k+1}^{-1})^{k+1} \ldots (\mathcal{A}_i^{-1})^i (\widehat{\widetilde{X}}^{t+1} - \widetilde{\mathcal{M}})$$

using the most recent estimates of the parameters. Assuming that the values of the parameter values are correct, we can write $\widetilde{Z} \sim \phi(\widetilde{Z}; \widetilde{0}, I_{m_1}, I_{m_2}, \ldots, I_{m_{k-1}}, \mathcal{A}_k, I_{m_{k+1}}, \ldots, I_{m_i})$, i.e., $Z_{(k)} \sim \phi(Z_{(k)}; \boldsymbol{0}_{m_k \times \prod_{j \neq k} m_j}, \mathcal{A}_k, I_{\prod_{j \neq k} m_j})$ where $Z_{(k)}$ denotes the $m_k \times \prod_{j \neq k} m_j$ matrix obtained by stacking the elements of $\widetilde{Z}$ along the kth dimension. Therefore, $(Z_{(k)1}, Z_{(k)2}, \ldots, Z_{(k)N}) = (\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots \boldsymbol{z}_{N \prod_{j \neq k} m_j})$ can be treated as a random

sample of size $N\prod_{j\neq k} m_j$ from the m_k-variate normal distribution with mean zero and covariance $\boldsymbol{\Sigma}_k = \mathcal{A}_k A_k'$. An update for $\boldsymbol{\Sigma}_k$ can be obtained by calculating:

$$\widehat{\boldsymbol{\Sigma}}_k^{t+1} = \frac{1}{N\prod_{j\neq k} m_j} \sum_{q=1}^{N\prod_{j\neq k} m_j} \left(Z_{(k)q} Z_{(k)q}' + \omega M_q' \left[R_q \boldsymbol{\Sigma}_k R_q' - R_q \boldsymbol{\Sigma}_k M_q' (M_q \boldsymbol{\Sigma}_k M_q')^{-1} M_q \boldsymbol{\Sigma}_k R_q'\right] M_q\right) \tag{5.6}$$

where R_q and M_q are the design matrices for the observed and missing values for the qth column of $(X_{(k)1}, \ldots, X_{(k)N})$, and $0 < \omega \approx 1$ is a hyperparameter.

The term

$$\frac{1}{N\prod_{j\neq k} m_j} \sum_{q=1}^{N\prod_{j\neq k} m_j} M_q' \left[R_q \boldsymbol{\Sigma}_k R_q' - R_q \boldsymbol{\Sigma}_k M_q' \left(M_q \boldsymbol{\Sigma}_k M_q'\right)^{-1} M_q \boldsymbol{\Sigma}_k R_q'\right] M_q$$

is the conditional expectation of

$$\frac{1}{N\prod_{j\neq k} m_j} \sum_{q=1}^{N\prod_{j\neq k} m_j} Z_{(k)q} Z_{(k)q}'$$

given the observed values and the current values of the parameters under the assumption that the location of the missing values in $(Z_{(k)1}, Z_{(k)2}, \ldots, Z_{(k)N})$ and $(X_{(k)1}, \ldots, X_{(k)N})$ is the same. This term is added to the usual covariance estimator to account for the fact that some cells are not observed but estimated quantities, and ω is used to account for the fact that the adjustment is not exact. In our applications, we have used $\omega = 1$, and it might be possible to improve the estimates for parameters by selecting a different value.

5.5 Flip-Flop Algorithm for Incomplete Arrays

For the incomplete matrix variate observations with Kronecker delta covariance structure, parameter estimation and missing data imputation methods have been developed in [2].

The following is a modification of the flip-flop algorithm for the incomplete array variable observations:

Algorithm 1 Given the current values of the parameters, repeat steps 1–3 until convergence:

1. Update $\widehat{\widehat{Y}}_i$ using (5.4),
2. Update $\widetilde{\mathcal{M}}$ using (5.5),
3. For $k = 1, 2, \ldots, i$, update $\boldsymbol{\Sigma}_k$ using (5.6).

5.6 A Semi-parametric Mixed-Effects Model

A semi-parametric mixed-effects model (SPMM) for the $n \times 1$ response vector $\boldsymbol{y}$ is expressed as

$$\boldsymbol{y} = X\beta + Z\boldsymbol{g} + \boldsymbol{e} \tag{5.7}$$

where $X\beta$ is the $n \times 1$ mean vector and Z is the $n \times q$ design matrix for the random effects; the random effects $(\boldsymbol{g}', \boldsymbol{e}')'$ are assumed to follow a multivariate normal distribution with mean $\mathbf{0}$ and covariance

$$\begin{pmatrix} \sigma_g^2 K & \mathbf{0} \\ \mathbf{0} & \sigma_e^2 I_n \end{pmatrix}$$

where K is a $q \times q$ kernel matrix. In general, the kernel matrix is a $k \times k$ non-negative definite matrix that measures the known degree of relationships between the k random effects. By the property of the multivariate normal distribution, the response vector $\boldsymbol{y}$ has a multivariate normal distribution with mean $X\beta$ and covariance $\sigma_g^2(ZKZ' + \lambda I)$ where $\lambda = \sigma_e^2/\sigma_g^2$.

The parameters of this model can be obtained maximizing the likelihood or the restricted likelihood (defined as the likelihood function with the fixed-effect parameters integrated out [9]). The estimators for the coefficients of the SPMM in (5.7) can be obtained via Henderson's iterative procedure. Bayesian procedures are discussed in detail in the book by Sorensen and Gianola [31]. An efficient likelihood-based algorithm (the efficient mixed-models association (EMMA)) was described in Kang et al. [20].

When there are more than one sources of variation acting upon the response vector $\boldsymbol{y}$, we may want to separate the influence of these sources. For such cases, we recommend using the following multiway random-effects model based on the multiway normal distribution in Definition 2.

Definition 6 A multiway random-effects model (AVSPMM) for the $m_1 \times m_2, \ldots \times m_i$ response array $\widetilde{Y}$ can be expressed as

$$\widetilde{Y} \sim \phi(\widetilde{Y}; \widetilde{\mathcal{M}(\boldsymbol{x})}, \sigma(K_1 + \lambda_1 I_{m_1})^{1/2}, (K_2 + \lambda_2 I_{m_2})^{1/2}, \ldots, (K_i + \lambda_i I_{m_i})^{1/2}) \tag{5.8}$$

where $\widetilde{M(\boldsymbol{x})}$ is an $m_1 \times m_2, \ldots \times m_i$ dimensional mean function of the observed fixed effects $\boldsymbol{x}$, and $K_1, K_2, \ldots, K_i$ are $m_1 \times m_1$, $m_2 \times m_2$, $\ldots$, $m_i \times m_i$ dimensional known kernel matrices measuring the similarity of the $m_1, m_2, \ldots, m_i$ levels of the random effects. If the covariance structure along the jth dimension is unknown, then the covariance along this dimension is assumed to be an unknown correlation matrix, i.e., we replace the term $(K_j + \lambda_j I_{m_j})$ by a single covariance matrix $\boldsymbol{\Sigma}_j$.

The parameter σ is arbitrarily associated with the first variance component and measures the total variance in the variable $\widetilde{Y}$ explained by the similarity matrices K_1, $K_2, \ldots, K_i$. λ_k represents the error-to-signal variance ratio along the kth dimension.

For the identifiability of the model, additional constraints on the covariance parameters are needed. Here, we adopt the restriction that the first diagonal element of the unknown covariance matrices is equal to one.

It is insightful to write the covariance structure for the vectorized form of the two-dimensional array model: In this case,

$$\begin{aligned} cov(rvec(\widetilde{Y})) &= \sigma^2(K_2 + \lambda_2 I_{m_1}) \otimes (K_1 + \lambda_1 I_{m_2}) \\ &= \sigma^2(K_2 \otimes K_1 + \lambda_1 K_2 \otimes I_{m_1} + \lambda_2 I_{m_2} \otimes K_1 + \lambda_1\lambda_2 I_{m_1 m_2}). \end{aligned} \tag{5.9}$$

If the covariance structure along the second dimension is unknown, then the model for the covariance of the response becomes

$$\begin{aligned} cov(rvec(\widetilde{Y})) &= \sigma^2(K_2 + \lambda_2 I_{m_1}) \otimes \boldsymbol{\Sigma}_2 \\ &= \sigma^2(\boldsymbol{\Sigma}_2 \otimes K_1 + \lambda_1 \boldsymbol{\Sigma}_2 \otimes I_{m_1}). \end{aligned} \tag{5.10}$$

It should be noted that the SPMM is related to the reproducing kernel Hilbert spaces (RKHS) regression so as the AVSPMM. The similarity of the kernel-based SPMMs and reproducing kernel Hilbert spaces (RKHS) regression models has been stressed recently [12]. In fact, this connection was previously recognized by [16, 21, 27, 32]. RKHS regression models use an implicit or explicit mapping of the input data into a high-dimensional feature space defined by a kernel function. This is often referred to as the "kernel trick" [30].

A kernel function, $k(.,.)$, maps a pair of input points $\boldsymbol{x}$ and $\boldsymbol{x}'$ into real numbers. It is by definition symmetric ($k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$) and non-negative. Given the inputs for the n individuals, we can compute a kernel matrix K whose entries are $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The linear kernel function is given by $k(\boldsymbol{x}; \boldsymbol{y}) = \boldsymbol{x}'\boldsymbol{y}$. The polynomial kernel function is given by $k(\boldsymbol{x}; \boldsymbol{y}) = (\boldsymbol{x}'\boldsymbol{y} + c)^d$ for c and $d \in R$. Finally, the Gaussian kernel function is given by $k(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{\sqrt{2\pi h}} exp(-(\boldsymbol{x}' - \boldsymbol{y})'(\boldsymbol{x}' - \boldsymbol{y})/2h)$ where $h > 0$. Taylor expansions of these kernel functions reveal that each of these kernels corresponds to a different feature map.

RKHS regression extends SPMMs by allowing a wide variety of kernel matrices, not necessarily additive in the input variables, calculated using a variety of kernel functions. The common choices for kernel functions are the linear, polynomial, Gaussian kernel functions, though many other options are available.

We also note that the AVSPMM is different than the standard multivariate mixed model for the matrix variate variables [17], in which the covariance for the vectorized form of the response vector is expressed as

$$cov(rvec(\widetilde{Y})) = (\boldsymbol{\Sigma}_{21} \otimes K_1 + \boldsymbol{\Sigma}_{22} \otimes I_{m_1}) \tag{5.11}$$

where $\boldsymbol{\Sigma}_{21}$ and $\boldsymbol{\Sigma}_{22}$ are m_2 dimensional unconstrained covariance matrices and the structure in (5.10) can be obtained by the restriction $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{22}$. A model in the sprit of the multivariate mixed model for the matrix variate variables can be extended to the array case by writing

$$\widetilde{Y} = \widetilde{M}(x) + \widetilde{G} + \widetilde{E} \tag{5.12}$$

where $\widetilde{M}(x)$ is the fixed-effects term and $\widetilde{G}$ and $\widetilde{E}$ are random-effects terms which are assumed to have independent array variate normal distributions, i.e., $\widetilde{G} \sim \phi(\mathbf{0}, K_1^{1/2}, \boldsymbol{\Sigma}_{21}^{1/2}, \boldsymbol{\Sigma}_{31}^{1/2}, \ldots, \boldsymbol{\Sigma}_{i1}^{1/2})$ and $\widetilde{E} \sim \phi(\mathbf{0}, I, \boldsymbol{\Sigma}_{22}^{1/2}, \boldsymbol{\Sigma}_{32}^{1/2}, \ldots, \boldsymbol{\Sigma}_{i2}^{1/2})$. It is straightforward to extend this last model to incorporate known relationship matrices along more than one dimension.

5.6.1 Models for the Mean

5.6.1.1 Model 1

A simple model for the mean is given by

$$\widetilde{M} = (\beta_1)^1 \mathbf{1}_{1 \times m_2 \times m_3 \times \ldots \times m_i} + (\beta_2)^2 \mathbf{1}_{m_1 \times 1 \times m_3 \times \ldots \times m_i} + \ldots + (\beta_i)^i \mathbf{1}_{m_1 \times m_2 \times m_3 \times \ldots \times 1} \tag{5.13}$$

where the $\beta_k \in \mathbf{R}^{m_k}$ for $k = 1, 2, \ldots, i$ are the coefficient vectors and the notation $\mathbf{1}_{m_1 \times m_2 \times m_3 \times \ldots \times m_i}$ refers to an $m_1 \times m_2 \times m_3 \times \ldots \times m_i$ dimensional array of ones. Note that this can also be written as

$$\begin{aligned} \widetilde{M} = {} & (\beta_1)^1 (\mathbf{1}_{m_2})^2 \ldots (\mathbf{1}_{m_k})^k 1 \\ & + (\mathbf{1}_{m1})^1 (\beta_2)^2 (\mathbf{1}_{m_3})^3 \ldots (\mathbf{1}_{m_k})^k 1 \\ & + \ldots + (\mathbf{1}_{m_1})^1 (\mathbf{1}_{m_2})^2 \ldots (\mathbf{1}_{m_{k-1}})^{k-1} (\beta_k)^k 1. \end{aligned}$$

Elementwise, this can be written as

$$(\widetilde{M})_{q_1 q_2 \ldots q_i} = (\beta_1)_{q_1} + (\beta_2)_{q_2} + \ldots + (\beta_i)_{q_i}.$$

This generalizes the model for the mean of two-dimensional arrays recommended in Allen and Tibshirani [2] to the general i dimensional case. For this model, the fixed-effects variables $\boldsymbol{x}$ are implicitly the effects of levels of the separable dimensions and some of which might be excluded by fixing the corresponding coefficient vector at zero during the modeling stage.

5.6.1.2 Model 2

If an explanatory variable in the form of an q dimensional vector $\boldsymbol{x}$ is observed along with each independent replication of the response variable, we can write a more general mixed model by modeling the mean with

$$\widetilde{M}(\boldsymbol{x}; B_1, \ldots, B_i) = (B_1)^1(\mathbf{1}_{m_2})^2 \ldots (\mathbf{1}_{m_i})^i \widetilde{\boldsymbol{x}^1} + (\mathbf{1}_{m_1})^1(B_2)^2(\mathbf{1}_{m_3})^3 \ldots (\mathbf{1}_{m_i})^i \widetilde{\boldsymbol{x}^2} + \ldots + (\mathbf{1}_{m_1})^1(\mathbf{1}_{m_2})^2 \ldots (\mathbf{1}_{m_{i-1}})^{i-1}(B_i)^i \widetilde{\boldsymbol{x}^i}. \quad (5.14)$$

where B_k is $m_k \times q$ for $k = 1, 2, \ldots, i$ and $\widetilde{\boldsymbol{x}^k}$ stands for the $1 \times \ldots \times 1 \times q \times 1 \times 1 \ldots \times 1$ dimensional array with q elements of $\boldsymbol{x}$ aligned along the kth dimension. This model encompasses the model for mean in (5.13). Elementwise, this can be expressed as

$$(\widetilde{M})_{q_1 q_2 \ldots q_i} = \sum_{p=1}^{q} (B_1)_{pq_1} + (B_2)_{pq_2} + \ldots + (B_i)_{pq_i} x_p.$$

At the modeling stage, some of B_k can be excluded from the model by fixing it at $\mathbf{0}$.

Let $\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_N$ be a random sample of array observations from the distribution with density $\phi(\widetilde{Y}; \widetilde{M}(\boldsymbol{x}; B_1, \ldots, B_i), \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i)$. Assuming that all parameters except B_k are known, the variable

$$\widetilde{Z}_\ell = (\widetilde{Y}_\ell - \widetilde{M}(\boldsymbol{x}_\ell; B_1, \ldots, B_{k-1}, B_k = 0, B_{k+1}, \ldots, B_i)$$

has density $\phi(\widetilde{Z}_\ell;\ \widetilde{M}(\boldsymbol{x}_\ell; B_1 = 0, \ldots, B_{k-1} = 0, B_k, B_{k+1} = 0, \ldots, B_i = 0),\ A_1, \ldots, A_i)$. Let $Z_{(k)\ell}$ denote the $m_k \times \prod_{j \neq k} m_j$ matrix obtained by matricization of $\widetilde{Z}_\ell$ along the kth dimension. $Z_{(k)\ell} = (\boldsymbol{z}_{1\ell}, \boldsymbol{z}_{2\ell}, \ldots \boldsymbol{z}_{\prod_{j \neq k} m_j \ell})$ has a matrix variate normal distribution with mean $B_k \boldsymbol{x}_\ell \mathbf{1}'_{\prod_{j \neq k} m_j}$ and covariances A_k and A_{-k} where $A_{-k} = A_i \otimes A_{i-1} \otimes \ldots \otimes A_{k-1} \otimes A_{k+1} \otimes \ldots \otimes A_1$. Let $Z^*_{(k)\ell} = Z_{(k)\ell} A^{-1}_{-k}$ and $X^*_{(k)\ell} = \boldsymbol{x}_\ell \mathbf{1}'_{\prod_{j \neq k} m_j} A^{-1}_{-k}$. Using the results that are already available for the multivariate regression [4], we can obtain the maximum-likelihood estimator of B_k;

$$\widehat{B_k} = \left[\sum_{\ell=1}^{N} Z^*_{(k)\ell} X^{*\prime}_{(k)\ell}\right] \left[\sum_{\ell=1}^{N} X^*_{(k)\ell} X^{*\prime}_{(k)\ell}\right]^{-1}. \quad (5.15)$$

5.6.1.3 Model 3

Let an explanatory variable in the form of a $1 \times \ldots \times m_j \times 1 \times \ldots \times 1 \times q$ dimensional array $\widetilde{X}$ in $i + 1$ dimensions be observed with each independent replication of the response variable. In this scenario, the fixed-effects variable $\widetilde{X}$ measures q variables that change along the jth dimension. We can write a more general mixed model by modeling the mean with

$$\begin{aligned}\widetilde{M}(\widetilde{X}_\ell;\{B_1,\ldots,B_i\}/B_j) &= (B_1)^1(\mathbf{1}_{m_2})^2\ldots(\mathbf{1}_{m_{j-1}})^{j-1}(I_{m_j})^j(\mathbf{1}_{m_{j+1}})^{j+1}\ldots(\mathbf{1}_{m_i})^i\widetilde{X}_\ell^1\\ &+(\mathbf{1}_{m_1})^1(B_2)^2(\mathbf{1}_{m_3})^3\ldots(\mathbf{1}_{m_{j-1}})^{j-1}(I_{m_j})^j(\mathbf{1}_{m_{j+1}})^{j+1}\ldots(\mathbf{1}_{m_i})^i\widetilde{X}_\ell^2\\ &+\ldots\\ &+(\mathbf{1}_{m_1})^1(\mathbf{1}_{m_2})^2(\mathbf{1}_{m_3})^3\ldots(B_{j-1})^{j-1}(I_{m_j})^j(\mathbf{1}_{m_{j+1}})^{j+1}\ldots(\mathbf{1}_{m_i})^i\widetilde{X}_\ell^{j-1}\\ &+(\mathbf{1}_{m_1})^1(\mathbf{1}_{m_2})^2(\mathbf{1}_{m_3})^3\ldots(\mathbf{1}_{m_{j-1}})^{j-1}(I_{m_j})^j(B_{j+1})^{j+1}\ldots(\mathbf{1}_{m_i})^i\widetilde{X}_\ell^{j+1}\\ &+\ldots\\ &+(\mathbf{1}_{m_1})^1(\mathbf{1}_{m_2})^2\ldots(\mathbf{1}_{m_{j-1}})^{j-1}(I_{m_j})^j(\mathbf{1}_{m_{j+1}})^{j+1}\ldots(\mathbf{1}_{m_{i-1}})^{i-1}(B_i)^i\widetilde{X}_\ell^i.\end{aligned} \tag{5.16}$$

where B_k is $m_k \times q$ for $k = 1, 2, \ldots, i$ and $\widetilde{X}_\ell^k$ stands for the $q \times \ldots \times m_j \times 1 \times \ldots \times 1$ dimensional array obtained by stacking $q \times \ldots \times 1 \times 1 \times \ldots \times 1$ arrays $\widetilde{\boldsymbol{x}_{\ell_c}^k}$ $c = 1, 2, \ldots, m_j$ along the jth dimension. Suppose $\widetilde{Y}$ is a three-dimensional response variable with dimensions $2 \times 2 \times m_3$ and $X = \left[\boldsymbol{x}_0 = \mathbf{1}_{m_3}, \boldsymbol{x}_1\right]$ is a $2 \times m_3$ dimensional explanatory variable which we can organize into a $1 \times 1 \times m_3 \times 2$ array $\widetilde{X}$. Then, $\widetilde{M}(\widetilde{X}; B_1, B_2)$ is $2 \times 2 \times m_3$ with the jth slice along the third dimension equal to

$$\begin{bmatrix}(B_1)_{11}+(B_2)_{11}+((B_1)_{12}+(B_2)_{12})X_{j2} & (B_1)_{11}+(B_2)_{21}+((B_1)_{12}+(B_2)_{22})X_{j2}\\ (B_1)_{21}+(B_2)_{11}+((B_1)_{22}+(B_2)_{12})X_{j2} & (B_1)_{21}+(B_2)_{21}+((B_1)_{22}+(B_2)_{22})X_{j2}\end{bmatrix}.$$

Let $\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_N$ be a random sample of array observations from the distribution with density $\phi(\widetilde{Y}; \widetilde{M}(\widetilde{X}; B_1, \ldots, B_i), \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i)$. Assuming that all parameters except B_k are known, the variable

$$\widetilde{Z}_\ell = (\widetilde{Y}_\ell - \widetilde{M}(\widetilde{X}_\ell; B_1, \ldots, B_{k-1}, B_k = 0, B_{k+1}, \ldots, B_i)$$

has density $\phi(\widetilde{Z}_\ell;\ \widetilde{M}(\boldsymbol{x}_\ell; B_1 = 0, \ldots, B_{k-1} = 0, B_k, B_{k+1} = 0, \ldots, B_i = 0), A_1, \ldots, A_i)$. Let $Z_{(k)\ell}$ denote the $m_k \times \prod_{j\neq k} m_j$ matrix obtained by matricization of $\widetilde{Z}_\ell$ along the kth dimension. $Z_{(k)\ell} = (\boldsymbol{z}_{1\ell}, \boldsymbol{z}_{2\ell}, \ldots \boldsymbol{z}_{\prod_{j\neq k} m_j \ell})$ has a matrix variate normal distribution with mean $B_k\widetilde{X}_\ell^k B_{-k}$ where $B_{-k} = \mathbf{1}'_{m_i} \otimes \mathbf{1}'_{m_{i-1}} \otimes \ldots \otimes \mathbf{1}'_{m_{k-1}} \otimes \mathbf{1}'_{m_{k+1}} \otimes \ldots \otimes \mathbf{1}'_{m_{j-1}} \otimes I_{m_j} \otimes \mathbf{1}'_{m_{j+1}} \otimes \ldots \otimes \mathbf{1}'_{m_1}$. Row and column covariances of $Z_{(k)\ell}$ are given by A_k and A_{-k} where $A_{-k} = A_i \otimes A_{i-1} \otimes \ldots \otimes A_{k-1} \otimes A_{k+1} \otimes \ldots \otimes A_1$. Let $Z^*_{(k)\ell} = Z_{(k)\ell}A^{-1}_{-k}$ and $X^*_{(k)\ell} = \widetilde{X}^k_\ell B_{-k}A^{-1}_{-k}$. The maximum-likelihood estimator of B_k is given by

$$\widehat{B_k} = \left[\sum_{\ell=1}^N Z^*_{(k)\ell}X^{*'}{}_{(k)\ell}\right]\left[\sum_{\ell=1}^N X^*_{(k)\ell}X^{*'}{}_{(k)\ell}\right]^{-1}. \tag{5.17}$$

$\widehat{B_k}$ is an unbiased estimator for B_k, and the covariance of it is given by

$$cov(vec(\widehat{B_k})) = \left[\sum_{\ell=1}^N X^*_{(k)\ell}X^{*'}{}_{(k)\ell}\right]^{-1} \otimes A_k.$$

A natural generalization of tests of significance of regression coefficients in univariate regression for B_k is

$$H_0 : \boldsymbol{l}_k' vec(B_k) = 0$$
$$H_1 : \boldsymbol{l}_k' vec(B_k) \neq 0.$$

A test statistic for testing this hypothesis is given by

$$T_k = (\boldsymbol{l}_k' vec(\widehat{B}_k))^2 / \boldsymbol{l}_k' cov(vec(\widehat{B}_k)) \boldsymbol{l}_k.$$

Under the null hypothesis, T_k has $\chi^2(1)$ distribution asymptotically.

Note that Model 1 and Model 2 are special cases of Model 3. Model 3 can be extended to incorporate input variables that change along more than one dimensions of the response variable.

5.6.1.4 Model 4

The generalization of the growth curve model to multiway data is obtained by considering the form

$$M(X_1, X_2, \ldots, X_i; \widetilde{B}) = (X_1)^1 (X_2)^2 \ldots (X_i)^i \widetilde{B}. \tag{5.18}$$

In (5.18), X_k for $k = 1, 2, \ldots, i$ are $m_k \times p_k$ known design matrices and $\widetilde{B}$ is the unknown parameter array of dimensions $p_1 \times p_2 \times \ldots \times p_i$. For example, if the levels of the kth dimension of a $m_1 \times m_2 \times \ldots \times m_i$ dimensional response variable $\widetilde{Y}$ is reserved for placing observations taken at points $\{x_{k1}, x_{k2}, \ldots, x_{km_i}\}$, X_k might be chosen as the design matrix of the $p_1 - 1$ degree monomials, i.e.,

$$X_k = \begin{bmatrix} 1 & x_{k1} & x_{k1}^2 & . & x_{k1}^{p_1-1} \\ 1 & x_{k2} & x_{k2}^2 & . & x_{k2}^{p_1-1} \\ . & . & . & . & . \\ 1 & x_{km_i} & x_{km_i}^2 & . & x_{km_i}^{p_1-1} \end{bmatrix}.$$

Let $\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_N$ be a random sample of array observations from the distribution with density $\phi(M(X_1, X_2, \ldots, X_i; \widetilde{B}), \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i)$. The density of the random sample $\widetilde{Y} = [\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_N]$ written in the form of a $m_1 \times m_2 \times \ldots \times m_i \times N$ array is $\phi(M(X_1, X_2, \ldots, X_i, \mathbf{1}_N; \widetilde{B}), \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i, I_N)$. Assuming that all parameters except $\widetilde{B}$ are known, the variable $\widetilde{Y^*} = (A_1^{-1})^1 \ (A_2^{-1})^2 \ldots (A_i^{-1})^i \ (I_N)^{i+1} \ \widetilde{Y}$ has a $\phi(M(A_1^{-1}X_1, A_2^{-1}X_2, \ldots, A_i^{-1}X_i, \mathbf{1}_N; \widetilde{B}), I_{m_1}, I_{m_2}, \ldots, I_{m_i}, I_N)$ distribution. Letting $X_1^* = A_1^{-1}X_1, X_2^* = A_2^{-1}X_2, \ldots, X_i^* = A_i^{-1}X_i$, the log-likelihood function is of the form

$$\begin{aligned}\ell(\widetilde{B}) \propto & -\left(rvec(\widetilde{Y^*}) - rvec(M(X_1^*, X_2^*, \ldots, X_i^*, \mathbf{1}_N; \widetilde{B}))\right)' \\ & \left(rvec(\widetilde{Y^*}) - rvec(M(X_1^*, X_2^*, \ldots, X_i^*, \mathbf{1}_N; \widetilde{B}))\right) \\ = & -\left(rvec(\widetilde{Y^*}) - \mathbf{1}_N \otimes X_i^* \otimes X_{i-1}^* \otimes \ldots \otimes X_1^* rvec(\widetilde{B}))\right)' \\ & \left(rvec(\widetilde{Y^*}) - \mathbf{1}_N \otimes X_i^* \otimes X_{i-1}^* \otimes \ldots \otimes X_1^* rvec(\widetilde{B}))\right) \\ = & -rvec(\widetilde{Y^*})' rvec(\widetilde{Y^*}) + 2rvec(\widetilde{Y^*})' \mathbf{1}_N \otimes X_i^* \otimes X_{i-1}^* \otimes \ldots \otimes X_1^* rvec(\widetilde{B}) \\ & -rvec(\widetilde{B})'(\mathbf{1}_N' \mathbf{1}_N \otimes X^{*\prime}{}_i X_i^* \otimes X^{*\prime}{}_{i-1} X_{i-1}^* \otimes \ldots \otimes X^{*\prime}{}_1 X_1^*) rvec(\widetilde{B})\end{aligned}$$

Taking the derivatives of $\ell(\widetilde{B})$ with respect to $rvec(\widetilde{B})$ and setting it to zero, we arrive at the normal equations

$$(\mathbf{1}_N \otimes X_i^* \otimes X_{i-1}^* \otimes \ldots \otimes X_1^*)' rvec(\widetilde{Y^*}) = (N \otimes X^{*\prime}{}_i X_i^* \otimes X^{*\prime}{}_{i-1} X_{i-1}^* \otimes \ldots \otimes X^{*\prime}{}_1 X_1^*) rvec(\widetilde{B}).$$

A solution of the normal equations can be expressed as

$$\widetilde{B} = ((X^{*\prime}{}_1 X_1^*)^{-1} X^{*\prime}{}_1)^1, (X^{*\prime}{}_2 X_2^*)^{-1} X^{*\prime}{}_2)^2, \ldots, (X^{*\prime}{}_i X_i^*)^{-1} X^{*\prime}{}_i)^i (\mathbf{1}_N'/N)^{i+1}) \widetilde{Y^*}. \tag{5.19}$$

Since the Hessian matrix

$$\frac{\partial^2 \ell(\widetilde{B})}{\partial rvec(\widetilde{B}) \partial rvec(\widetilde{B})'} = -(N \otimes X^{*\prime}{}_i X_i^* \otimes X^{*\prime}{}_{i-1} X_{i-1}^* \otimes \ldots \otimes X^{*\prime}{}_1 X_1^*)$$

is negative definite, $\widehat{\widetilde{B}}$ maximizes the log-likelihood function. Note that $\widehat{\widetilde{B}}$ is a linear function of $\widetilde{Y}$ and also has also normal distribution given by

$$\phi(\widetilde{B}, (X^{*\prime}{}_1 X_1^*)^{-1/2}, (X^{*\prime}{}_2 X_2^*)^{-1/2}, \ldots (X^{*\prime}{}_i X_i^*)^{-1/2}, 1/\sqrt{N}).$$

A test of significance for regression coefficient $\widetilde{B}$ is

$$\begin{aligned} H_0 &: \boldsymbol{l}' rvec(\widetilde{B}) = 0 \\ H_1 &: \boldsymbol{l}' rvec(\widetilde{B}) \neq 0. \end{aligned}$$

A test statistic for testing this hypothesis is given by

$$T = (\boldsymbol{l}' rvec(\widehat{\widetilde{B}}))^2 / \boldsymbol{l}' cov(rvec(\widehat{\widetilde{B}}))\boldsymbol{l}.$$

Under the null hypothesis, T has $\chi^2(1)$ distribution asymptotically.

5.6.1.5 Model 5

If an explanatory variable $\widetilde{X}$ of dimensions $p_1 \times p_2 \times \ldots \times p_i$ is observed with the $m_1 \times m_2 \times \ldots \times m_i$ dimensional response $\widetilde{Y}$, a model for the mean is written as

$$M(\widetilde{X}; B_1, B_1, \ldots, B_i) = (B_1)^1 (B_2)^2 \ldots (B_i)^i \widetilde{X}$$

where B_j is $m_j \times p_j$ for $j = 1, 2, \ldots, i$. Estimators of B_j can be obtained with an iterative algorithm similar to the flip-flop algorithm in Chap. 4.

5.6.2 *Models for the Covariance*

We can model the covariance structure along each dimension independently. Several models for the covariances are mentioned below.

5.6.2.1 Model 1

We can assume that A_k is unstructured. If all parameters except A_k are known, the maximum-likelihood estimator of A_k under the unstructured covariance assumption is given by (5.6).

5.6.2.2 Model 2

Now, we turn our attention to estimation of the covariance parameters $\{\sigma^2, \lambda_k\}$ for $k = 1, 2, \ldots, i$. Assume that the mean and all variance parameters other than $\{\sigma^2, \lambda_k\}$ are known. By standardizing the centered array variable in all but the kth dimension followed by matricization along the same dimension and finally vectorization (denote this $n^* = N \prod_{j=1}^{i} m_j$ vector by $z_{(k)}$), we obtain a multivariate mixed model for which estimates for $\{\sigma^2, \lambda_k\}$ can be obtained efficiently by using a slight modification of EMMA algorithm [20]. The distribution of the $z_{(k)}$ is

$$\phi_{N \prod_{j=1}^{i} m_j}(\mathbf{0}, \sigma^2 (I_{N \prod_{j \neq k} m_j} \otimes K_k + \lambda_k I)).$$

Let $H_k = (I_{N \prod_{j \neq k} m_j} \otimes K_k + \lambda_k I)$. The likelihood function is optimized at

$$\widehat{\sigma^2} = \frac{z'_{(k)} H_k^{-1} z_{(k)}}{N \prod_{j=1}^{i} m_j}$$

for fixed values of λ_k. Using the spectral decomposition of $H_k = U diag(\epsilon_1 + \lambda_k, \epsilon_2 + \lambda_k, \ldots, \epsilon_{N \prod_{j=1}^{i} m_j} + \lambda_k) U'$ and letting $\eta = U'\mathbf{y}$, the log-likelihood function for λ_k at $\widehat{\sigma^2}$ can be written as

$$l(\lambda) = \frac{1}{2}\left[-n^* log\frac{2\pi z'_{(k)} H_k^{-1} z_{(k)}}{n^*} - log|H_k| - n^*\right]$$
$$= \frac{1}{2}\left[n^* log\frac{n^*}{2\pi} - n^* - n^* log(\sum_{i=1}^{n^*}\frac{\eta_i^2}{\epsilon_i + \lambda_k}) - \sum_{i=1}^{n^*} log(\epsilon_i + \lambda_k)\right] \quad (5.20)$$

which can be maximized using univariate optimization. An additional efficiency is obtained by considering the singular value decomposition of a Kronecker product:

$$A \otimes B = (U_A D_A V'_A) \otimes (U_B D_B V'_B) = (U_A \otimes U_B)(D_A \otimes D_B)(V_A \otimes V_B)'.$$

That is, the left and right singular vectors and the singular values are obtained as Kronecker products of the corresponding matrices of the components. Therefore, we can calculate the eigenvalue decomposition of H_k efficiently using

$$H_k = (I \otimes U_k)(I \otimes (D_k + \lambda_k I))(I \otimes U_k)' \quad (5.21)$$

where $U_k(D_k + \lambda I)U'_k$ is the eigenvalue decomposition of $K_k + \lambda_k I$ and $U_k D U'_k$ is the eigenvalue decomposition of K_k.

Note that our estimation procedure for σ and λ_k ignores that fact that the response variable z_k is composed partially of estimated quantities.

5.6.2.3 Model 3

If there are two sources of inputs along a dimension of an array resulting in two kernel matrices K_1 and K_2, then a simple model for the covariance parameter along that dimension is given by considering a combination of these matrices and a product term

$$w_1 K_1 + w_2 K_2 + w_3 K_1 \odot K_2$$

where the "$\odot$" stands for the Hadamard product operator, $w_j \geq 0$ for $j = 1, 2, 3$ and $\sum_{j=1}^{3} w_j = 1$. It is easy to extend this idea to more than two sources of input, and a rich family of models is possible by considering only subsets of these terms.

5.6.2.4 Other Models

Consider the following covariance model for the vectorized form of a $m_1 \times m_2$ dimensional array $\widetilde{Y}$:

$$cov(rvec(\widetilde{Y})) = \sigma^2(K_2 \otimes K_1 + \lambda_1 K_2 \otimes I_{m_1} + \lambda_2 I_{m_2} \otimes K_1 + \lambda_3 I_{m_1 m_2}).$$

Since $\lambda_3 = \lambda_1 * \lambda_2$ is not imposed, the array model for the array $\widetilde{Y}$ cannot be expressed as in Definition 6. The model parameters can be estimated, for example,

using maximum likelihood. However, the estimation is computationally demanding since the efficiencies due to the Kronecker delta covariance structure are not available here.

5.6.3 A Flip-Flop Algorithm for Estimating the AVSPMM

Algorithm 1 can be adopted for the AVSPMM as follows:

Algorithm 2 Given the current values of the parameters, repeat steps 1–3 until convergence:

1. Update $\widehat{\widetilde{Y}}_\ell$ using (5.4) for $\ell = 1, 2, \ldots, N$.
2. For $k = 1, 2, \ldots, i$ update mean parameters along the kth dimension.
3. For $k = 1, 2, \ldots, i$ update covariance parameters along the kth dimension.

5.7 Principal Component Analysis for Array Variate Random Variables

Principal component analysis (PCA) is a useful statistical technique. The end products of PCA are a set of new set of uncorrelated variables (principal components) obtained from a linear combination of the original variables and the corresponding variances for these variables. It is customary to list the principal components in decreasing order of their variances.

Definition 7 For the $m_1 \times m_2 \times \ldots \times m_i$ dimensional array variate random variable $\widetilde{Y}$, the principal components are defined as the principal components of the $d = m_1 m_2 \ldots m_i$-dimensional random vector $rvec(\widetilde{Y})$.

For an array normal random variable $\widetilde{Y}$ with $E(rvec(\widetilde{Y})) = \mathbf{0}$ and covariance $cov(rvec(\widetilde{Y})) = \Lambda$, the principal components can be obtained by considering the eigenvalue decomposition of the covariance, $\Lambda = UDU'$. The columns of U are called the principal components. And $cov(U' rvec(\widetilde{Y})) = D$ is diagonal and the jth diagonal element of D corresponds to the variance of the random variable $U_j' rvec(\widetilde{Y})$.

When a random sample $\widetilde{Y}_1, \widetilde{Y}_2 \ldots \widetilde{Y}_N$ is observed and Λ is unknown, the principal components are calculated using the sample covariance of $rvec(\widetilde{Y}_1)$, $rvec(\widetilde{Y}_2) \ldots$ $rvec(\widetilde{Y}_N)$ instead of Λ since for $N > \prod_{k=1}^{i} m_i$ the sample covariance matrix is a consistent estimator of the covariance parameter. However, for high-dimensional arrays, usually $N < \prod_{k=1}^{i} m_i$, and the sample covariance is not a consistent estimator of Λ since it has at least one zero eigenvalue, whereas the parameter Λ is positive definite.

If we assume that the variable $\widetilde{Y}$ has a Kronecker delta covariance structure, i.e., $\widetilde{Y} \sim \phi(\widetilde{\mathbf{0}}, \mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i)$, then $\{\lambda(A_r)_{r_j}\}$ are the m_j eigenvalues of $A_r A_r'$ with

the corresponding eigenvectors $\{(\boldsymbol{x}_r)_{r_j}\}$ for $r = 1, 2, \ldots, i$ and $r_j = 1, 2, \ldots, m_r$, and then $\Lambda = (A_1 A_1' \otimes A_2 A_2' \otimes^i A_i A_i')$ will have eigenvalues $\{\lambda(A_1)_{r_1}\lambda(A_2)_{r_2} \ldots \lambda(A_i)_{r_i}\}$ with corresponding eigenvectors $\{(\boldsymbol{x}_i)_{r_i} \otimes (\boldsymbol{x}_2)_{r_2} \otimes \ldots \otimes (\boldsymbol{x}_i)_{r_i}\}$.

If the covariance parameters $\mathcal{A}_1, \mathcal{A}_2, \ldots \mathcal{A}_i$ are unknown, we can obtain sample-based estimates of them when $N \prod_{j \neq k} m_j > m_k$ (assuming there are no missing cells) using Algorithm 1. When covariance components along some of the dimensions are assumed to be known, the criterion for the sample size is further relaxed. We can estimate the eigenvalues and eigenvectors of the covariance of $rvec(\widetilde{X})$ by replacing the parameters by their estimators.

5.8 Classification

Suppose the array variable is generated by a mixture of two densities, i.e., $\widetilde{X} \sim \pi\ N(A_1,\ A_2,\ \ldots,\ A_i,\ M_1) + (1 - \pi)\ N(B_1,\ B_2,\ \ldots,\ B_i,\ M_2)$. Based on a set of training observations, the classification of a new observation to the component density $N(A_1, A_2, \ldots, A_i, M_1)$ or $N(B_1, B_2, \ldots, B_i, M_2)$ can be done using the Bayes rule using estimators of the parameters.

Let the training estimates of parameters be $\widehat{\pi}$, $\widehat{A_1}$, $\widehat{A_2}$, $\ldots$, $\widehat{A_i}$, $\widehat{M_1}$, $\widehat{B_1}$, $\widehat{B_2}$, $\ldots$, $\widehat{B_i}$, $\widehat{M_2}$. Then, the posterior probability of an observation with value $\widetilde{X}$ to come from the first component $N(A_1, A_2, \ldots, A_i, M_1)$ is given by

$$\widehat{P}(\widetilde{X} \sim N(A_1, A_2, \ldots, A_i, M_1)) = \frac{\widehat{\pi}\phi(\widetilde{X}; \widehat{A_1}, \widehat{A_2}, \ldots, \widehat{A_i}, \widehat{M_1})}{\widehat{\pi}\phi(\widetilde{X}; \widehat{A_1}, \widehat{A_2}, \ldots, \widehat{A_i}, \widehat{M_1}) + (1 - \widehat{\pi})\phi(\widetilde{X}; \widehat{B_1}, \widehat{B_2}, \ldots, \widehat{B_i}, \widehat{M_2})}. \tag{5.22}$$

According to this, we would classify the observation $\widetilde{X}$ to the first component if the posterior probability is large (e.g., if the probability is more than 0.5); otherwise, we classify it to the second component. The extension to the case of more than two component densities is straightforward.

5.9 Illustrations

Three real and two simulated data sets are used in this section to illustrate our models. These examples also serve to show the effects of changing sample size, missing data proportion, and array dimensions on the performance of methods.

Example 5.9.1 This data set was first given in [11] and was also studied in [10]. The data consist of skull length and breadth measured on siblings in 25 families. For each replication of this experiment, we randomly select n families from the 25 families available in the data set resulting in a $n \times 4$ data matrix. This matrix is centered by subtracting the column means from each row of the data. Next, we introduce missingness at random by deleting each cell of this matrix with probability p. The resulting

data are then imputed using three different models. The first model (m1) is obtained by assuming $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n \overset{iid}{\sim} \phi(\mathbf{0}, \boldsymbol{\Sigma}^{1/2})$ where $\boldsymbol{\Sigma}$ is a 4×4 unstructured positive definite covariance matrix. The standard EM algorithm is used for the inference and imputation. The second model (m2) assumes $\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_n \overset{iid}{\sim} \phi(\mathbf{0}, A_1, A_2)$ where the 2×2 dimensional arrays $\widetilde{Y}_i$ for $i = 1, 2, \ldots, n$ are obtained by organizing $\boldsymbol{y}_i$ into an array with first dimension for the siblings and the second dimension for the skull length and breadth measurements and the positive definite matrices A_1 and A_2 are the corresponding covariance parameters. The third model (m3) uses the genetic information that full-siblings in a randomly mating population are expected to be 50 % similar, i.e., we assume $\widetilde{Y}_1, \widetilde{Y}_2, \ldots, \widetilde{Y}_n \overset{iid}{\sim} \phi(\mathbf{0}, \sigma(K + \lambda I)^{1/2}, A_2)$ where $\sigma, \lambda > 0$, K is given by

$$\begin{bmatrix} 1 & 0.50 \\ 0.50 & 1 \end{bmatrix}.$$

For $n = 10, 15, 20, 25$ and $p = 0.5, 0.3, 0.2,$ we have calculated the correlation between the imputed and measured values with each model for 30 replications (30 replications where all three models gave solutions) and the results are summarized in Fig. 5.1. In addition, the estimates of the four eigenvalues (variances of the four principal components) of the covariance of $\boldsymbol{y}$ based on the three models for $n = 10, 15, 20, 25$ and $p = 0.5, 0.3, 0.2$ over 30 replications are summarized in Fig. 5.2. In all settings, the accuracies are higher (and the standard errors are lower) for the models with Kronecker delta covariance structure (m2, m3) compared to the accuracies for the model with unstructured covariance (m1). The accuracies from model m1 were the least robust to the increase in missing cells and small samples. Estimates of eigenvalues based on these models differed slightly.

Example 5.9.2 In this example, we have generated a random sample of $10 \times 4 \times 2$ dimensional array random variables according to a known array variate distribution. After that, we have randomly deleted a given proportion of the cells of these arrays. The algorithm for estimation 1 was implemented to estimate the parameters and to impute the missing cells. Finally, the correlation between the observed values of the missing cells and the imputed values and the mean-squared error (MSE) of the estimates of the overall Kronecker structured covariance matrix is calculated. We have tried sample sizes of 20, 50, and 100 and the missing data proportions of 0.4, 0.3, 0.2, and 0.1. The correlations and the MSEs were calculated for 30 independent replications, and these results are presented in Fig. 5.3. As expected, the solutions from our methods improve as the sample size increases or when the proportion of missing cells decreases.

Example 5.9.3 In an experiment conducted in Aberdeen during 2013, 524 barley lines from the North American Small Grain Collection were grown using combinations of two experimental factors. The levels of the first factor were the low and

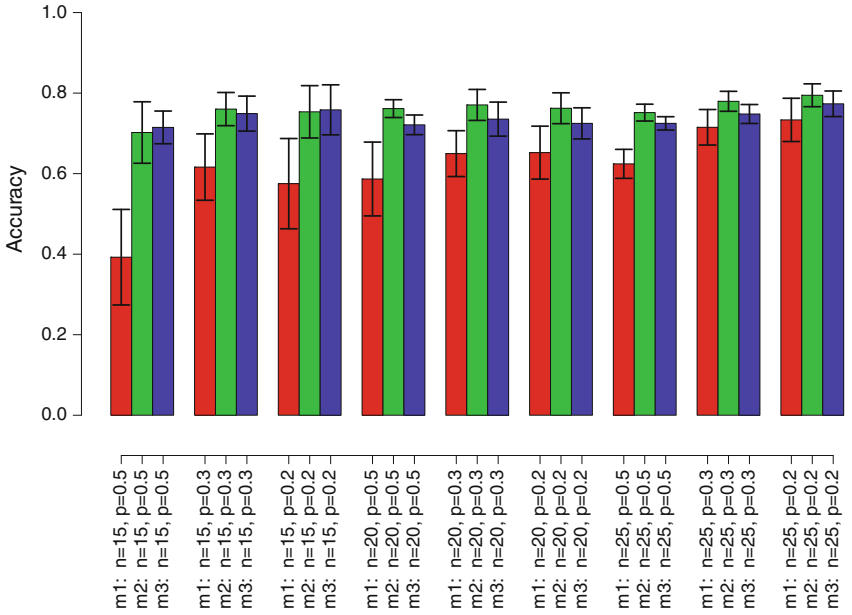


Fig. 5.1 The accuracies for the scenario in Example 1 summarized with the barplots and error bars for standard errors. In all settings, the accuracies are better for the models with Kronecker delta covariance structure (*m2*, *m3*) compared to the accuracies for the model with unstructured covariance (*m1*)

normal nitrogen, and the levels of the second experimental factor were dry and irrigated conditions. The low nitrogen and irrigation combination was not reported. Five traits, i.e., plant height, test weight, yield, whole-grain protein, and heading date (Julian), were used here. We have constructed an incomplete array of dimensions $524 \times 2 \times 2 \times 5$ from these data and induced additional missingness by randomly selecting a proportion $(0.6, 0.4, 0.1)$ of the cells and deleting the recorded values in these cells (regardless of whether the cell was already missing). In addition, 4803 SNP markers were available for all of the 524 lines which allowed us to calculate the covariance structure along this dimension, and the covariance structure along the other dimensions was assumed unknown. An additive mean structure for the means of different traits was used, and all the other mean parameters related to the other dimensions were assumed to be zero. For each trait, the correlation between the observed and the corresponding estimated values was calculated for 30 independent replications of this experiment with differing proportion of missing values, and these are summarized in Fig. 5.4. The results indicate that our methods provide a means

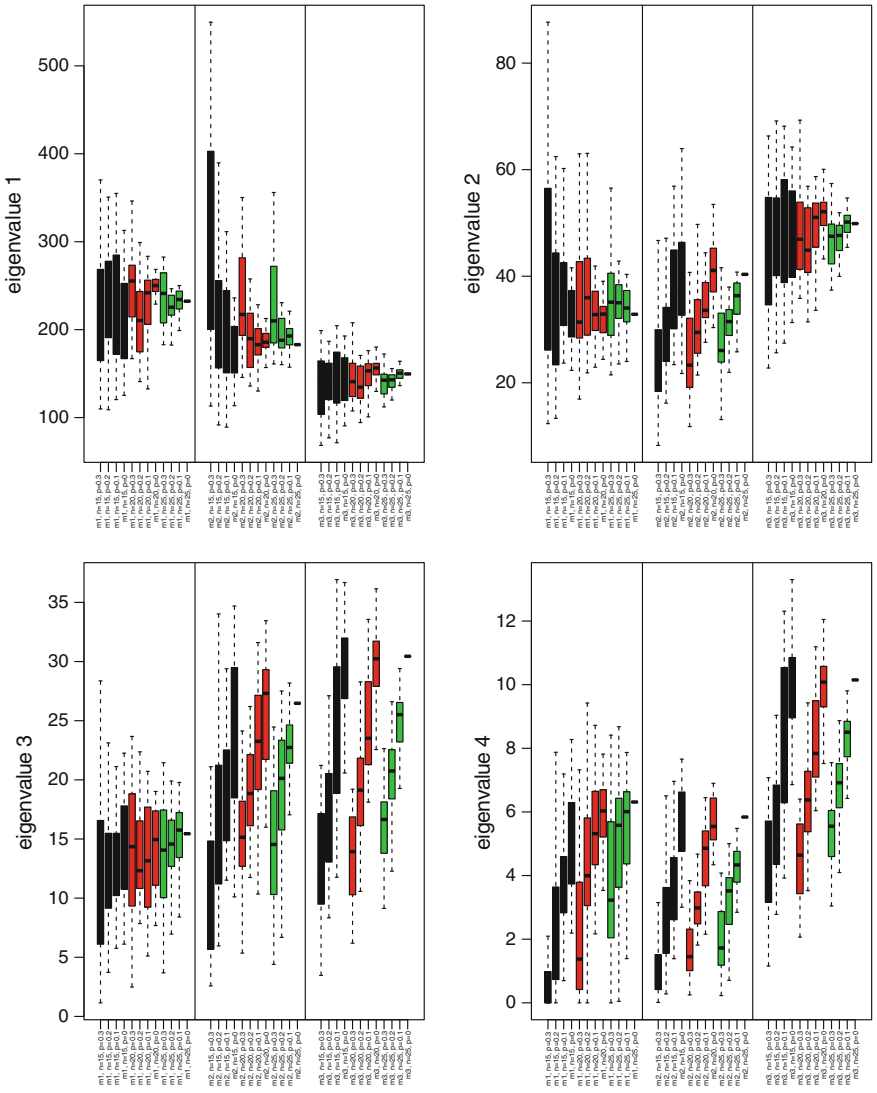


Fig. 5.2 The eigenvalues for the scenario in Example 1 summarized with the boxplots. Estimates of eigenvalues based on models *m1*, *m2*, and *m3* differed slightly

to estimate the traits that were generated by the combined effect of genetics and environment.

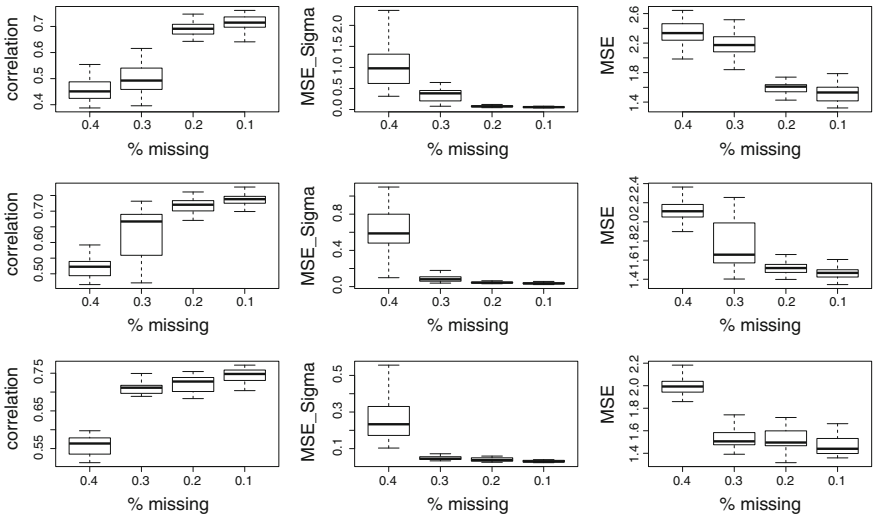


Fig. 5.3 The boxplots of the correlations (*left*) and the MSEs (*right*) for varying values of the sample size and missing cell proportions. As expected, the solutions from our methods improve as the sample size increases (*top* to *bottom*) or when the proportion of missing cells decreases (*left* to *right*)

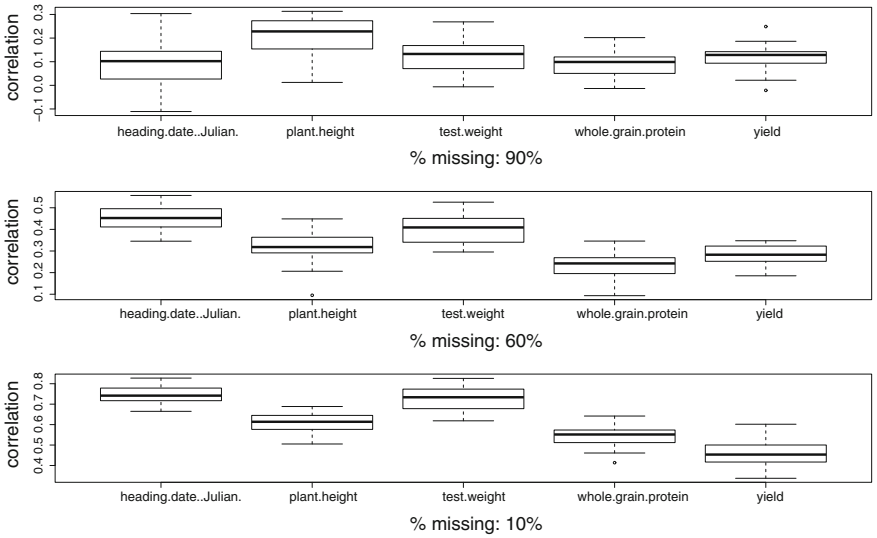


Fig. 5.4 The accuracies for the scenario in Example 2 summarized with the boxplots. The number of missing cells is highest for the bottom figure and lowest for the *top* figure

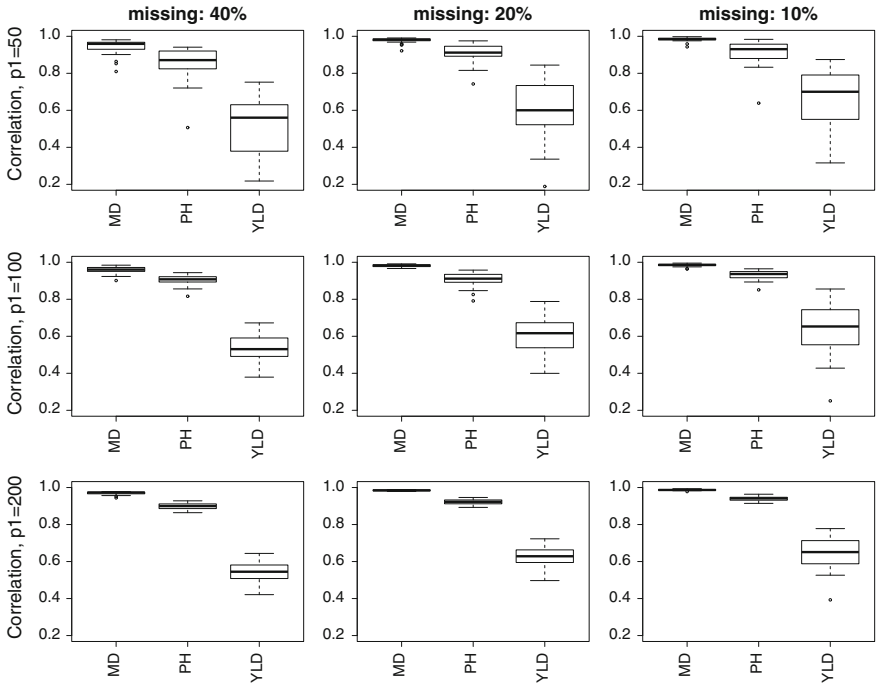


Fig. 5.5 The accuracies for the scenario in Example 3 summarized with the boxplots. The number of missing cells decreases from *left* to *right*, and p_1 increases from *top* to *bottom*

Example 5.9.4 In this example, we have used the data from an experiment conducted over two years. 365 lines from the spring wheat association mapping panel were each observed for three agronomical traits (plant height, yield, and physiological maturity date) in two separate year/location combinations under the irrigated and dry conditions. A 365×365 relationship matrix was obtained using 3735 SNP markers in the same fashion as Example 2. However, since we wanted to study the effect of the number of different genotypes on the accuracies, we have selected a random sample of p_1 genotypes out of the 365 where p_1 was taken as one of 50, 100, 200. The phenotypic data were used to form a $p_1 \times 2 \times 2 \times 3$ array. The entry in each cell was deleted with probabilities 0.4, 0.2 and 0.1. Finally, within-trait correlations between the missing cells and the corresponding estimates from the AVSPMM over 30 replications of each of the settings of this experiment are summarized by the boxplots in Fig. 5.5.

In addition, for a subset of 147 lines with complete yield measurements in both years and under both experimental settings, we have calculated the test statistic associated with the effects of 227 markers on the 3B chromosome in each of the four conditions by using Model 3 for the mean, fitted with a marker at a time. We have also obtained the results from a standard univariate EMMA-based association results obtained from a model that used the environments and a marker as fixed effects and a

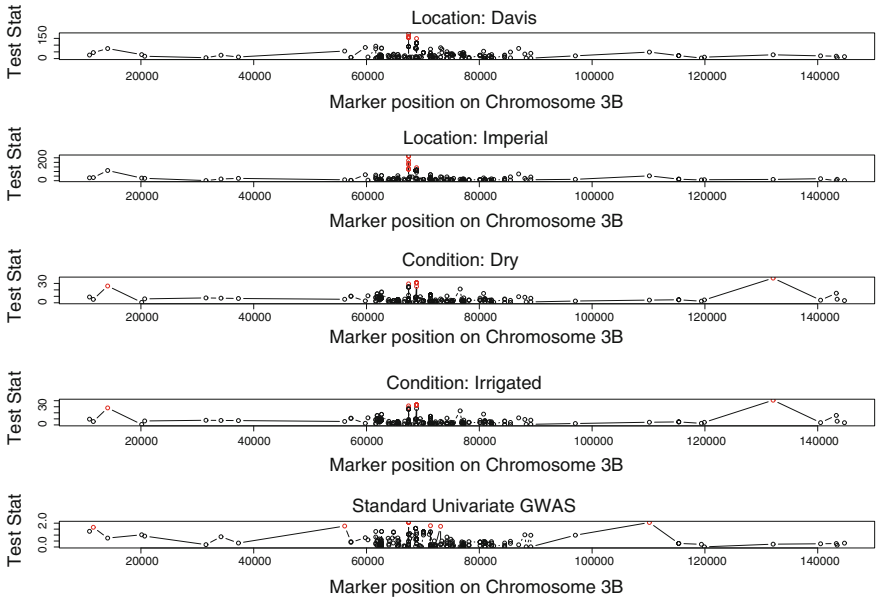


Fig. 5.6 Effects of 227 markers on the 3B chromosome in each of the four conditions by using Model 3 for the mean, and the trait is yield

genomewide kinship matrix for control of population structure used as the covariance structure for the lines. These results are summarized in Fig. 5.6. In general, the effects follow the same pattern in each of the four environments. There are regions close to both ends of the chromosome that have large effects in experimental conditions (dry, irrigated) and not in locations (Davis-Imperial), pointing to markers that respond to experimental conditions. The association results for different environments from the array model were slightly different than corresponding standard univariate results.

Example 5.9.5 These data involve simulations from a known AVSPMM model for a $p_1 \times 6 \times 2$ array, sample size 1. We demonstrate that the MSE for the overall covariance decreases with increasing p_1 where p_1 stands for the number of levels of the dimension for which the covariance structure is available in the estimation process. After generating the array variate response, we have deleted cells with probability 0.4, 0.2, or 0.1. This was replicated 30 times. The correlations and MSE between the estimated response and the corresponding known (but missing) cells and the MSE between the estimated and the known covariance parameters are displayed in Fig. 5.7.

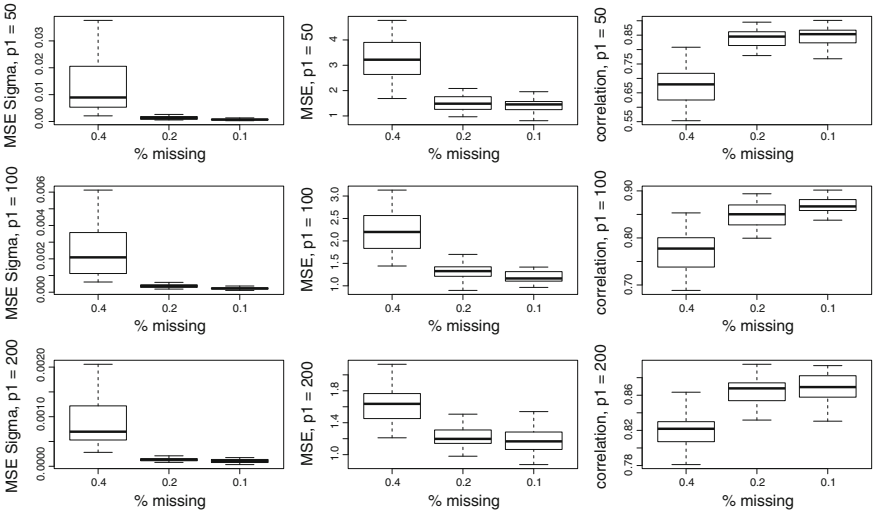


Fig. 5.7 The figures on the left display the MSE between the estimated and the known covariance parameters, and the figures on the right display the correlations between the estimated response and the corresponding known (but missing) cells for $p_1 = 50, 100, 200$ increasing downward and probability of missingness 0.4, 0.2, 0.1 decreasing toward the right

5.10 Discussion

We have formulated a parametric model for array variate data and developed suitable estimation methods for the parameters of this distribution with possibly incomplete observations. The main application of this paper has been to multiway regression (missing data imputation), and once the model parameters are given, we are able to estimate the unobserved components of any array from the observed parts of the array. We have assumed no structure on the missingness pattern; however, we have not explored the estimability conditions.

The proposed estimation algorithms can in some cases exhibit extremely slow convergence, and it did not always converge to a solution when the percentage of missing values in the array was large. In addition to large percentage of missing values, some other possible reasons for non-convergence include poor model specification, the missingness pattern, small sample size, and poor initial values for the parameters. In some of the instances of non-convergence, it might be possible to obtain convergence by combining the levels of one or more dimensions and decreasing the order of the array.

Extensions of the AVSPMM are possible by considering other models for the mean and the covariance parameters. Another possible model for the mean array can be obtained by the rank-R decomposition of the mean array parallel factors (PARAFAC) [7, 14] where an array is approximated by a sum of R rank one arrays. For a general ith order array of dimensions, $m_1 \times m_2, \ldots \times m_i$ rank-R decomposition

can be written as

$$\widetilde{M} = \sum_{k=1}^{R} \rho_k \mu_{r1} \circ \mu_{r2} \circ \ldots \circ \mu_{ri}$$

where $\mu_{rk} \in \mathbf{R}^{m_k}$ and $||\mu_{rk}||^2 = 1$ for $k = 1, 2, \ldots, i$. This can be written as

$$(\widetilde{M})_{q_1 q_2 \ldots q_i} = \sum_{k=1}^{R} \rho_k \mu_{r1 q_1} \mu_{r2 q_2} \ldots \mu_{r i q_i}.$$

The AVSPMM is a suitable model when the response variable is transposable and allows us to separate the variance in the array variate response into components along its dimensions. This model also allows us to make predictions for the unobserved level combinations of the dimensions as long as we know the relationship of these new levels to the partially observed levels along each separate dimension.

Acknowledgments This research was supported by the USDA-NIFA-AFRI Triticeae Coordinated Agricultural Project, award number 2011-68002-30029.

References

1. Akdemir, D., & Gupta, A. K. (2011). Array variate random variables with multiway kronecker delta covariance matrix structure. *Journal of Algebraic Statistics*, *2*(1), 98–113.
2. Allen, G. I., & Tibshirani, R. (2010). Transposable regularized covariance models with an application to missing data imputation. *The Annals of Applied Statistics*, *4*(2), 764–790.
3. Anderson, T. W. (1957). Maximum likelihood estimates for a multivariate normal distribution when some observations are missing. *Journal of the American Statistical Association*, *52*(278), 200–203.
4. Anderson, T. W. (1984). *An introduction to multivariate*. Wiley.
5. Beale, E. M. L., & Little, R. J. A. (1975). Missing values in multivariate analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 129–145.
6. Blaha, G. (1977). A few basic principles and techniques of array algebra. *Journal of Geodesy*, *51*(3), 177–202.
7. Bro, R. (1997). Parafac. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, *38*(2), 149–171.
8. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1–38.
9. Dempster, A. P., Rubin, D. B., & Tsutakawa, R. K. (1981). Estimation in covariance components models. *Journal of the American Statistical Association*, *76*(374), 341–353.
10. Flury, B. (1997). *A first course in multivariate statistics*. Springer.
11. Pieter, G. F. (1921). Heredity of headform in man. *Genetica*, *3*(3), 193–400.
12. Gianola, D., & Van Kaam, J. B. (2008). Reproducing kernel hilbert spaces regression methods for genomic assisted prediction of quantitative traits. *Genetics*, *178*(4), 2289–2303.
13. Gupta, A. K., & Nagar, D. K. (2000). *Matrix variate distributions*. In: Chapman and Hall/CRC Monographs and Surveys in Pure and Applied Mathematics. London: Chapman and Hall.
14. Harshman, R. A. (1970). Foundations of the parafac procedure: models and conditions for an "explanatory" multimodal factor analysis. UCLA Working Papers in Phonetics.

15. Hartley, H. O., & Hocking, R.R. (1971). The analysis of incomplete data. *Biometrics*, 783–823.
16. Harville, D. A. (1983). Discussion on a section on interpolation and estimation. In: DHA and HT David (ed.), *Statistics an Appraisal* (pp. 281–286). Ames: The Iowa State University Press.
17. Henderson, C. R., & Quaas, R. L. (1976). Multiple trait evaluation using relatives' records. *Journal of Animal Science*, *43*(6), 1188–1197.
18. Hoff, P. D. (2011). Hierarchical multilinear models for multiway data. *Computational Statistics and Data Analysis*, *55*(1), 530–543.
19. Jørgensen, B., & Petersen, H. C. (2012). Efficient estimation for incomplete multivariate data. *Journal of Statistical Planning and Inference*, *142*(5), 1215–1224.
20. Kang, H. M., Zaitlen, N. A., Wade, C. M., Kirby, A., Heckerman, D., Daly, M. J., et al. (2008). Efficient control of population structure in model organism association mapping. *Genetics*, *178*(3), 1709–1723.
21. Kimeldorf, G. S., & Wahba, G. (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 495–502.
22. Lu, N., & Zimmerman, D. L. (2005). The likelihood ratio test for a separable covariance matrix. *Statistics and Probability Letters*, *73*(4), 449–457.
23. Meng, X. L., & Rubin, D. B. (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, *80*(2), 267–278.
24. Ohlson, M., Ahmad, M. R., & von Rosen D. (2011). The multilinear normal distribution: Introduction and some basic properties. *Journal of Multivariate Analysis*.
25. Orchard, T., & Woodbury, M. A. (1972). A missing information principle: theory and applications. In: *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability* (vol. 1, pp. 697–715).
26. Rauhala, U. A. (1974). *Array Algebra with Applications in Photogrammetry and Geodesy*. Division of Photogrammetry, Royal Institute of Technology.
27. Robinson, G. K. (1991). That blup is a good thing: The estimation of random effects. *Statistical Science*, *6*(1), 15–32.
28. Roy, A., & Khattree, R. (2003). Tests for mean and covariance structures relevant in repeated measures based discriminant analysis. *Journal of Applied Statistical Science*, *12*(2), 91–104.
29. Roy, A., & Leiva, R. (2008). Likelihood ratio tests for triply multivariate data with structured correlation on spatial repeated measurements. *Statistics and Probability Letters*, *78*(13), 1971–1980.
30. Schölkopf, B., & Smola, A. (2005). *Learning with Kernels*. Cambridge: MIT Press.
31. Sorensen, D., & Gianola, D. (2002). *Likelihood, Bayesian, and MCMC methods in quantitative genetics*. Springer.
32. Speed, T. (1991). Comment on "That blup is a good thing: The estimation of random effects". *Statistical Science*, *6*(1), 42–44.
33. Srivastava, M. S., Nahtman, T., & Von Rosen, D. (2008). *Estimation in general multivariate linear models with kronecker product covariance structure*. Report: Research Report Centre of Biostochastics, Swedish University of Agriculture science. 1.
34. Srivastava, M. S., Von Rosen, T., & Von Rosen, D. (2008). Models with a kronecker product covariance structure: Estimation and testing. *Mathematical Methods of Statistics*, *17*(4), 357–370.
35. Trawinski, I. M., & Bargmann, R. E. (1964). Maximum likelihood estimation with incomplete multivariate data. *The Annals of Mathematical Statistics*, *35*(2), 647–657.

Chapter 6
One-Sided Tests for Matrix Variate Normal Distribution

Manabu Iwasa and Toshio Sakata

Abstract In this chapter, we consider one-sided tests of the mean matrix M for a matrix variate normal distribution $N_{p,q}(M, \Sigma, \Psi)$. When Σ is unknown and Ψ is known, we construct a class of similar test statistics. The class includes similar test statistics obtained by Sasabuchi [8] and Hu and Banerjee [2].

Keywords Matrix variate normal distribution · One-sided test · Similar test · Inequality constraints

6.1 Introduction

The study of one-sided test under an order restriction for several means of one-dimensional normal distributions was originated by Bartholomew [1], and it was generalized to multivariate normal distributions by Kudo [4]. Then, they were concerned mainly with the likelihood ratio tests(LRTs) under a known covariance structure in a multivariate setting. See, for example, Robertson et al. [7] and Silvapulle and Sen [11].

In general, the covariance structure is unknown in multivariate analysis, and Perlman [6] discussed one-sided testing problems in a multivariate normal distribution with completely unknown covariance matrix. He derived the LRT and showed that there exist similar tests which are uniformly more powerful than the LRT. Sasabuchi, Tanaka and Tsukamoto [9] and Sasabuchi [8] studied order-restricted testing problems for mean vectors of several multivariate normal distributions with a common

M. Iwasa (✉)
Kumamoto University, 2-39-1 Kurokami Chuou-ku, Kumamoto 860-8555, Japan
e-mail: iwasa@gpo.kumamoto-u.ac.jp

T. Sakata
Department of Human Science of Kyushu University, 4-9-1 Siobaru Minami-ku, Fukuka, Japan
e-mail: sakata.toshio.923@m.kyushu-u.ac.jp

T. Sakata (ed.), *Applied Matrix and Tensor Variate Data Analysis*,
JSS Research Series in Statistics, DOI 10.1007/978-4-431-55387-8_6

unknown covariance matrix. In this setting of several normal distributions, it is difficult to derive the LRT statistic, and thus, they introduced a test statistic other than the LRT statistic. Furthermore, they showed that there exist similar test statistics which are more powerful than the proposed test statistic. Hu and Banerjee [2] developed the similar argument for more general order restrictions.

One-sided testing problem in a matrix (and tensor) variate normal distribution has not been discussed in any literature. A matrix variate normal distribution $N_{p,q}(M, \Sigma, \Psi)$ has a density function

$$f(X, M, \Sigma, \Psi) = (2\pi)^{-\frac{pq}{2}} |\Sigma|^{-\frac{q}{2}} |\Psi|^{-\frac{p}{2}} \exp\left\{\frac{1}{2}\mathrm{tr}\Sigma^{-1}(X - M)\Psi^{-1}(X - M)'\right\}, \tag{6.1}$$

where $M = (\mu_{ij})$ is a $p \times q$ mean matrix and Σ and Ψ are $p \times p$ and $k \times k$ positive definite matrices, respectively. It is well known that a matrix variate normal distribution $N_{p,q}(M, \Sigma, \Psi)$ is rewritten as a vector variate normal distribution $N_{pq}(\mathrm{vec}(M), \Psi \otimes \Sigma)$ by vectorization. See Kollo and von Rosen [3] for a theory of matrix variate normal distributions. Srivastava et al. [12] discussed statistical inferences in general multivariate linear model for matrix variate normal distributions. Our interest is in one-sided testing problems for matrix variate normal distributions and is to extend results in Sasabuchi [8] and Hu and Banerjee [2].

In Sect. 6.2, we give preliminary results on a linear space of matrices. We summarize some properties concerning the projection onto a cone in the space of matrices. In Sect. 6.3, we briefly discuss the LRT when both Σ and Ψ are known. In Sect. 6.4, we consider the case where Σ is unknown and Ψ is known. We present a class of similar test statistics which is wider than those by Sasabuchi [8] and Hu and Banerjee [2].

6.2 Preliminaries

Throughout this chapter, for any vectors $\boldsymbol{x}$, $\boldsymbol{y}$ we consider the standard inner product and the induced norm defined by $\boldsymbol{x}'\boldsymbol{y}$ and $||\boldsymbol{x}|| = \sqrt{\boldsymbol{x}'\boldsymbol{x}}$, respectively. On the other hand, we introduce a following extended inner product and norm for $p \times q$ matrices. $\mathbb{R}^{p,q}$ denotes the linear space of all $p \times q$ matrices with real valued elements.

Definition 6.2.1 For positive definite matrices $G \in \mathbb{R}^{p,p}$ and $H \in \mathbb{R}^{q,q}$, we define an inner product $(X, Y)_{G,H}$ and a norm of $||X||_{G,H}$ on $\mathbb{R}^{p,q}$ as follows.

$$(X, Y)_{G,H} = \mathrm{tr}(G^{-1}XH^{-1}Y'), \quad ||X||_{G,H} = \sqrt{(X, X)_{G,H}}.$$

The following lemma is obvious from Definition 6.2.1.

Lemma 6.2.1 *Suppose $G, \Gamma \in \mathbb{R}^{p\cdot p}$ are positive definite and $H, \Lambda \in \mathbb{R}^{q,q}$ are nonsingular. Then, for any $A, B \in \mathbb{R}^{p,q}$, it holds that*

$$\begin{aligned}(A, B)_{\Gamma G\Gamma', \Lambda' H\Lambda} &= (\Gamma^{-1}A\Lambda^{-1}, \Gamma^{-1}B\Lambda^{-1})_{G,H},\\ ||A||_{\Gamma G\Gamma', \Lambda' H\Lambda} &= ||\Gamma^{-1}A\Lambda^{-1}||_{G,H}.\end{aligned} \tag{6.2}$$

For a positive definite matrix A, $A^{1/2}$ denotes a symmetric square root of A and $A^{-1/2}$ denotes the inverse of $A^{1/2}$. The density function (6.1) is written as

$$f(X, M, \Sigma, \Psi) = (2\pi)^{-\frac{pq}{2}}|\Sigma|^{-\frac{q}{2}}|\Psi|^{-\frac{p}{2}}\exp\{-\frac{1}{2}||X - M||^2_{\Sigma,\Psi}\}. \tag{6.3}$$

Definition 6.2.2 For $A \in \mathbb{R}^{p,q}$ and a closed convex set $\mathcal{R} \subset \mathbb{R}^{p,q}$, $\Pi_{G,H}(A|\mathcal{R})$ denotes the matrix $B \in \mathcal{R}$ minimizing $||A - B||_{G,H}$. That is, $\Pi_{G,H}(A|\mathcal{R}) = \arg\min_{B\in\mathcal{R}} ||A - B||_{G,H}$.

$\Pi_{G,H}(A|R)$ is called the projection of A onto $\mathcal{R}$ with respect to $||\cdot||_{G,H}$. Since $\mathcal{R} \subset \mathbb{R}^{p,q}$ is closed and convex, $\Pi_{G,H}(A|R)$ is determined uniquely. For a set $\mathcal{R} \subset \mathbb{R}^{p,q}$, we define

$$\Gamma\mathcal{R}\Lambda = \{\Gamma B\Lambda \in \mathbb{R}^{p,q} \mid B \in \mathcal{R}\}.$$

When $\mathcal{R}$ is closed and convex, $\Gamma\mathcal{R}\Lambda$ is also closed and convex.

Lemma 6.2.2 *When $\Gamma \in \mathbb{R}^{p,p}$ and $\Lambda \in \mathbb{R}^{q,q}$ are nonsingular, for any $A \in \mathbb{R}^{p,q}$ and a closed convex set $\mathcal{R} \subset \mathbb{R}^{p,q}$, we have*

$$\begin{aligned}&\Pi_{G,H}(\Gamma A\Lambda|\Gamma\mathcal{R}\Lambda) = \Gamma\{\Pi_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{R})\}\Lambda,\\ &||\Pi_{G,H}(\Gamma A\Lambda|\Gamma\mathcal{R}\Lambda)||_{G,H} = ||\Pi_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{R})||_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}.\end{aligned}$$

Proof From Definition 6.2.1 and Lemma 6.2.1, we have

$$\begin{aligned}\Pi_{G,H}(\Gamma A\Lambda|\Gamma\mathcal{R}\Lambda) &= \arg\min_{B^*\in\Gamma\mathcal{R}\Lambda} ||\Gamma A\Lambda - B^*||_{G,H}\\ &= \Gamma(\arg\min_{B\in\mathcal{R}} ||\Gamma A\Lambda - \Gamma B\Lambda||_{G,H})\Lambda\\ &= \Gamma(\arg\min_{B\in\mathcal{R}} ||A - B||_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}})\Lambda\\ &= \Gamma\Pi_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{R})\Lambda,\end{aligned}$$

and

$$\begin{aligned}&||\Pi_{G,H}(\Gamma A\Lambda|\Gamma\mathcal{R}\Lambda)||_{G,H}\\ &= ||\Gamma\Pi_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{R})\Lambda||_{G,H}\\ &= ||\Pi_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{R})||_{\Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}}.\end{aligned}$$

□

In one-sided testing problems, we consider hypotheses constrained by linear equalities and inequalities. Let V and W be an $s \times p$ matrix and a $q \times t$ matrix, respectively. We define a subspace $\mathcal{L}(V, W)$ and a convex cone $\mathcal{C}(V, W)$ on $\mathbb{R}^{p,q}$ as follows.

$$\mathcal{L}(V, W) = \{M \in \mathbb{R}^{p,q} \mid VMW = O\},$$
$$\mathcal{C}(V, W) = \{M \in \mathbb{R}^{p,q} \mid VMW \geq O\},$$

where O is an $s \times t$ zero matrix and the inequality $VMW \geq O$ implies that all elements of VMW are nonnegative. Throughout this chapter, we consider the case where the convex cone $\mathcal{C}(V, W)$ is not degenerate; that is, $\mathcal{C}(V, W)$ has interior points.

Moreover, we define subspaces and cones in vector spaces $\mathbb{R}^p$ and $\mathbb{R}^q$ as follows.

$$\mathcal{L}(V, \bullet) = \{\boldsymbol{x} \in \mathbb{R}^p \mid V\boldsymbol{x} = 0\}, \ \ \mathcal{C}(V, \bullet) = \{\boldsymbol{x} \in \mathbb{R}^p \mid V\boldsymbol{x} \geq 0\},$$
$$\mathcal{L}(\bullet, W) = \{\boldsymbol{y} \in \mathbb{R}^q \mid \boldsymbol{y}'W = 0\}, \ \ \mathcal{C}(\bullet, W) = \{\boldsymbol{y} \in \mathbb{R}^q \mid \boldsymbol{y}'W \geq 0\}.$$

Lemma 6.2.3 *If $\boldsymbol{x} \in \mathbb{R}^p$ and $\boldsymbol{y} \in \mathbb{R}^q$ are interior points of $\mathcal{C}(V, \bullet)$ and $\mathcal{C}(\bullet, W)$ respectively, $\boldsymbol{x}\boldsymbol{y}' \in \mathbb{R}^{p,q}$ is an interior point of $\mathcal{C}(V, W)$.*

Proof Since $V\boldsymbol{x} > 0$ and $\boldsymbol{y}'W > 0$ by the assumptions, we have $V(\boldsymbol{x}\boldsymbol{y}')W = (V\boldsymbol{x})(\boldsymbol{y}'W) > O$. This implies that $\boldsymbol{x}\boldsymbol{y}'$ is interior points of $\mathcal{C}(V, W)$. □

We give two important examples of $\mathcal{C}(V, W)$ and $\mathcal{L}(V, W)$.

Example 6.2.1 Sasabuchi et al. [9] discussed testing homogeneity of multivariate normal mean vectors under simple ordering. The cone discussed by them is represented by

$$C = \{M = (\mu_{ij}) \mid \mu_{i1} \leq \mu_{i2} \leq \cdots \leq \mu_{iq}, \ i = 1, 2, \ldots, p\}.$$

In our formulation, $C = \mathcal{C}(V, W)$ for

$$V = I_p \ (s = p), \ \ W = \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 \\ 1 & -1 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix} \ (t = q - 1).$$

When $V = I_p$, each row of M has an identical constraint induced by W, and such constraints are said to be synchronized by Hu and Banerjee [2]. They studied one-sided tests under general synchronized constraints. The next example gives a non-synchronized constraint.

Example 6.2.2 Set $s = p - 1, t = q - 1$ and

$$V = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \\ 0 & 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{pmatrix}, \quad W = \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 \\ 1 & -1 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -1 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Then, we have

$$\mathcal{C}(V, W) = \{M = (\mu_{ij}) \mid \mu_{i'j} + \mu_{ij'} \le \mu_{ij} + \mu_{i'j'},\ 1 \le i < i' \le p; 1 \le j < j' \le q\}.$$

When $M \in \mathcal{L}(V, W)$, M is additive; that is, μ_{ij} is decomposed as $\mu_{ij} = \mu + \alpha_i + \beta_j$. On the other hand, when $M \in \mathcal{C}(V, W)$, all submatrices $\begin{pmatrix} \mu_{ij} & \mu_{ij'} \\ \mu_{i'j} & \mu_{i'j'} \end{pmatrix}$ have nonnegative interactions in the sense that $\mu_{ij} + \mu_{i'j'} - \mu_{i'j} - \mu_{ij'} \ge 0$.

When $\Gamma \in \mathbb{R}^{p,p}$ and $\Lambda \in \mathbb{R}^{q,q}$ are nonsingular, we have

$$\begin{aligned} \Gamma\mathcal{L}(V, W)\Lambda &= \{\Gamma A \Lambda \mid VAW = O\} \\ &= \{\Gamma A \Lambda \mid V\Gamma^{-1}\Gamma A \Lambda \Lambda^{-1} W = O\} = \mathcal{L}(V\Gamma^{-1}, \Lambda^{-1}W). \end{aligned}$$

Thus, we obtain the next lemma.

Lemma 6.2.4 *If $\Gamma \in \mathbb{R}^{p,p}$ and $\Lambda \in \mathbb{R}^{q,q}$ are nonsingular, it holds that*

$$\Gamma\mathcal{L}(V, W)\Lambda = \mathcal{L}(V\Gamma^{-1}, \Lambda^{-1}W),\ \Gamma\mathcal{C}(V, W)\Lambda = \mathcal{C}(V\Gamma^{-1}, \Lambda^{-1}W).$$

Lemma 6.2.5 *When $\Gamma \in \mathbb{R}^{p,p}$ is nonsingular, we have the following.*
(i) $\mathcal{L}(V, \bullet) = \mathcal{L}(V\Gamma, \bullet)$ implies $\mathcal{L}(V\Gamma, W) = \mathcal{L}(V, W)$.
(ii) $\mathcal{C}(V, \bullet) = \mathcal{C}(V\Gamma, \bullet)$ implies $\mathcal{C}(V\Gamma, W) = \mathcal{C}(V, W)$.

Proof Let $\boldsymbol{w}_i$ be the ith column vector of W, that is, $W = (\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_t)$. Then, since $VAW = (VA\boldsymbol{w}_1, VA\boldsymbol{w}_2, \ldots, VA\boldsymbol{w}_t)$, if $\mathcal{L}(V, \bullet) = \mathcal{L}(V\Gamma, \bullet)$, we obtain the following.

$$\begin{aligned} & A \in \mathcal{L}(V, W) = \{A \in \mathbb{R}^{p,q} \mid VAW = O\} \\ \Longleftrightarrow\ & A\boldsymbol{w}_i \in \mathcal{L}(V, \bullet) \quad (\forall i = 1, \ldots, t) \\ \Longleftrightarrow\ & A\boldsymbol{w}_i \in \mathcal{L}(V\Gamma, \bullet) \quad (\forall i = 1, \ldots, t) \\ \Longleftrightarrow\ & A \in \mathcal{L}(V\Gamma, W) = \{A \in \mathbb{R}^{p,q} \mid V\Gamma AW = O\}. \end{aligned}$$

This completes the proof of (i). Proof of (ii) is proved similarly. □

Let $\boldsymbol{v}_i$ be the transpose of the ith row of V, that is, $V = (\boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots\ \boldsymbol{v}_s)'$. Then we define a linear space $\mathcal{L}\langle V\rangle$ and a convex cone $\mathcal{C}\langle V\rangle$ in $\mathbb{R}^p$ by

$$\mathcal{L}\langle V\rangle = \{c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_s\boldsymbol{v}_s \mid c_i \in \mathbb{R}\ (i = 1, 2, \ldots, s)\},$$
$$\mathcal{C}\langle V\rangle = \{c_1\boldsymbol{v}_1 + c_2\boldsymbol{v}_2 + \cdots + c_s\boldsymbol{v}_s \mid c_i \geq 0\ (i = 1, 2, \ldots, s)\}.$$

Note that $\mathcal{L}\langle V\rangle$ is the orthogonal complement of $\mathcal{L}(V, \bullet)$ and $\mathcal{C}\langle V\rangle$ is a dual cone of $\mathcal{C}(V, \bullet)$.

Lemma 6.2.6 *For any $\boldsymbol{u} \in \mathcal{C}\langle V\rangle$, we have $\mathcal{C}(V, W) \subset \mathcal{C}(\boldsymbol{u}', W)$.*

Proof Suppose that $A \in \mathcal{C}(V, W)$. Then it holds that $VAW \geq O$, which implies that $\boldsymbol{v}_i'AW \geq 0$ for all $i = 1, 2, \ldots, s$. Since $\boldsymbol{u} = c_1\boldsymbol{v}_1 + \cdots + c_s\boldsymbol{v}_s$ for $c_i \geq 0$ $(i = 1, 2, \ldots, s)$, we have $\boldsymbol{u}'AW = \sum_{i=1}^{s} c_i(\boldsymbol{v}_i'AW) \geq 0$, which implies that $A \in \mathcal{C}(\boldsymbol{u}', W)$. □

For any fixed $\boldsymbol{u} \in \mathcal{C}\langle V\rangle$, let $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p\}$ (where $\boldsymbol{u}_1 = \dfrac{\boldsymbol{u}}{||\boldsymbol{u}||}$) be an orthonormal basis of $\mathbb{R}^p$. We define a sequence of nonsingular matrices Ξ_n by

$$\Xi_n = \Delta_n\Delta_1^{-1} \qquad (n = 1, 2, \ldots) \tag{6.4}$$

where Δ_n $(n = 1, 2, \ldots)$ is a nonsingular matrix defined by $(n\boldsymbol{u}_1\ \boldsymbol{u}_2\ \cdots\ \boldsymbol{u}_p)$. Note that $\Delta_1^{-1} = \Delta_1'$ because Δ_1 is an orthogonal matrix.

Theorem 6.2.1 *Suppose that $\boldsymbol{u} \in \mathcal{C}\langle V\rangle$ satisfies $\boldsymbol{v}_i'\boldsymbol{u} > 0$ for all $i = 1, 2, \ldots, s$. Then, for Ξ_n defined by (6.4), we have*

$$\mathcal{C}(V\Xi_1, W) \subset \mathcal{C}(V\Xi_2, W) \subset \cdots \subset \mathcal{C}(V\Xi_n, W) \subset \cdots \tag{6.5}$$

and the closure of $\bigcup_{n=1}^{\infty} \mathcal{C}(V\Xi_n, W)$ is equal to $\mathcal{C}(\boldsymbol{u}', W)$.

Proof Noting that $\mathcal{C}(V, W) \subset \mathcal{C}(\boldsymbol{u}', W)$ and $\mathcal{C}(V, W) = \bigcap_{i=1}^{s} \mathcal{C}(\boldsymbol{v}_i', W)$, we have

$$\begin{aligned}\mathcal{C}(V, W) &= \mathcal{C}(V, W) \cap \mathcal{C}(\boldsymbol{u}', W) \\ &= \{\bigcap_{i=1}^{s} \mathcal{C}(\boldsymbol{v}_i', W)\} \cap \mathcal{C}(\boldsymbol{u}', W) = \bigcap_{i=1}^{s} \{\mathcal{C}(\boldsymbol{v}_i', W) \cap \mathcal{C}(\boldsymbol{u}', W)\}.\end{aligned}$$

Putting $\boldsymbol{v}_i = \sum_{j=1}^{p} \alpha_{ij}\boldsymbol{u}_j$ for the orthonormal basis $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p\}$, we have

$$\begin{aligned}\boldsymbol{v}_i'\Xi_n &= \boldsymbol{v}_i'\Delta_n\Delta_1^{-1} = \boldsymbol{v}_i'(n\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p)(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p)' \\ &= n\alpha_{i1}\boldsymbol{u}_1' + \sum_{j=2}^{p} \alpha_{ij}\boldsymbol{u}_j'.\end{aligned} \tag{6.6}$$

If $A \in \mathcal{C}(\boldsymbol{u}', W)$ (i.e., $\boldsymbol{u}_1'AW \geq 0$), it holds that for all $n = 2, 3, \ldots$

$$\begin{aligned}\boldsymbol{v}_i'\Xi_n AW &= n\alpha_{i1}\boldsymbol{u}_1'AW + \sum_{j=2}^{p}\alpha_{ij}\boldsymbol{u}_j'AW \\ &= \alpha_{i1}\boldsymbol{u}_1'AW + \boldsymbol{v}_i'\Xi_{n-1}AW \ \geq \ \boldsymbol{v}_i'\Xi_{n-1}AW,\end{aligned}$$

because $\alpha_{i1} = \boldsymbol{v}_i'\boldsymbol{u}_1 > 0$. Thus, for all $i = 1, 2, \ldots, s$, we have

$$\begin{aligned}\mathcal{C}(\boldsymbol{v}_i'\Xi_1, W) \cap \mathcal{C}(\boldsymbol{u}', W) &\subset \mathcal{C}(\boldsymbol{v}_i'\Xi_2, W) \cap \mathcal{C}(\boldsymbol{u}', W) \\ &\subset \cdots \subset \mathcal{C}(\boldsymbol{v}_i'\Xi_n, W) \cap \mathcal{C}(\boldsymbol{u}', W) \subset \cdots . \end{aligned} \tag{6.7}$$

On the other hand, since

$$\boldsymbol{u}'\Xi_n = \boldsymbol{u}'(n\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p)(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_p)' = n\boldsymbol{u}',$$

we have $\mathcal{C}(\boldsymbol{u}'\Xi_n, W) = \mathcal{C}(\boldsymbol{u}', W)$. Therefore, we obtain (6.5) from (6.7) because

$$\begin{aligned}\mathcal{C}(V\Xi_n, W) &= \Xi_n^{-1}\mathcal{C}(V, W) = \Xi_n^{-1}\{\mathcal{C}(V, W) \cap \mathcal{C}(\boldsymbol{u}', W)\} \\ &= \Xi_n^{-1}\{\bigcap_{i=1}^{s}\mathcal{C}(\boldsymbol{v}_i', W) \cap \mathcal{C}(\boldsymbol{u}', W)\} \\ &= \Xi_n^{-1}\bigcap_{i=1}^{s}\{\mathcal{C}(\boldsymbol{v}_i', W) \cap \mathcal{C}(\boldsymbol{u}', W)\} \\ &= \bigcap_{i=1}^{s}\{\mathcal{C}(\boldsymbol{v}_i'\Xi_n, W) \cap \mathcal{C}(\boldsymbol{u}', W)\}.\end{aligned}$$

Since (6.6) implies that $\frac{1}{n}\boldsymbol{v}_i'\Xi_n \to \alpha_{i1}\boldsymbol{u}_1'$ as $n \to \infty$, we have $\mathcal{C}(\boldsymbol{v}_i'\Xi_n, W) \to \mathcal{C}(\boldsymbol{u}', W)$ as $n \to \infty$ for all $i = 1, 2, \ldots, s$. Thus, the second statement follows. □

We present several properties of the projection Π onto $\mathcal{L}(V, W)$ and $\mathcal{C}(V, W)$.

Lemma 6.2.7 *For any $A \in \mathbb{R}^{p,q}$ and $B \in \mathcal{L}(V, W)$, we have*
(i) $\Pi_{G,H}(A + B|\mathcal{L}(V, W)) = \Pi_{G,H}(A|\mathcal{L}(V, W)) + B$,
(ii) $\Pi_{G,H}(A + B|\mathcal{C}(V, W)) = \Pi_{G,H}(A|\mathcal{C}(V, W)) + B$.

Lemma 6.2.8 *For any $A \in \mathbb{R}^{p,q}$, we have*
(i) $||A||_{G,H}^2 = ||\Pi_{G,H}(A|\mathcal{L}(V, W))||_{G,H}^2 + ||A - \Pi_{G,H}(A|\mathcal{L}(V, W))||_{G,H}^2$,
(ii) $||A||_{G,H}^2 = ||\Pi_{G,H}(A|\mathcal{C}(V, W))||_{G,H}^2 + ||A - \Pi_{G,H}(A|\mathcal{C}(V, W))||_{G,H}^2$,
(iii) $||A - \Pi_{G,H}(A|\mathcal{L}(V, W))||_{G,H}^2 - ||A - \Pi_{G,H}(A|\mathcal{C}(V, W))||_{G,H}^2$
$= ||\Pi_{G,H}(A|\mathcal{C}(V, W)) - \Pi_{G,H}(A|\mathcal{L}(V, W))||_{G,H}^2$.

For $A \in \mathbb{R}^{p,q}$, $G \in \mathbb{R}^{p,p}$, $H \in \mathbb{R}^{q,q}$, $V \in \mathbb{R}^{s,p}$, $W \in \mathbb{R}^{q,t}$, we define

$$T_{V,W}(A, G, H) = ||\Pi_{G,H}(A|\mathcal{C}(V, W)) - \Pi_{G,H}(A|\mathcal{L}(V, W))||_{G,H}^2. \tag{6.8}$$

Lemma 6.2.9 *For any $A \in \mathbb{R}^{p,q}$, we have the following statements.*
(i) $T_{V,W}(A+B,G,H) = T_{V,W}(A,G,H)$ if $B \in \mathcal{L}(V,W)$.
(ii) $T_{V,W}(\Gamma A \Lambda, G, H) = T_{V\Gamma,\Lambda W}(A, \Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1})$ if $\Gamma \in \mathbb{R}^{p,p}$ and $\Lambda \in \mathbb{R}^{q,q}$ are nonsingular.

Proof (i) It is obvious from Lemma 6.2.7 and (6.8). (ii) By Lemmas 6.2.1, 6.2.2 and 6.2.4, we have

$$\begin{aligned}
&T_{V,W}(\Gamma A \Lambda, G, H) \\
=&||\Pi_{G,H}(\Gamma A \Lambda|\mathcal{C}(V,W)) - \Pi_{G,H}(\Gamma A \Lambda|\mathcal{L}(V,W))||^2_{G,H} \\
=&||\Gamma\{\Pi_{\Gamma^{-1}G\Gamma'^{-1},\Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{C}(V\Gamma,\Lambda W)) \\
&- \Pi_{\Gamma^{-1}G\Gamma'^{-1},\Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{L}(V\Gamma,\Lambda W))\}\Lambda||^2_{G,H} \\
=&||\Pi_{\Gamma^{-1}G\Gamma'^{-1},\Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{C}(V\Gamma,\Lambda W)) \\
&- \Pi_{\Gamma^{-1}G\Gamma'^{-1},\Lambda'^{-1}H\Lambda^{-1}}(A|\mathcal{L}(V\Gamma,\Lambda W))||^2_{\Gamma^{-1}G\Gamma'^{-1},\Lambda'^{-1}H\Lambda^{-1}} \\
=&T_{V\Gamma,\Lambda W}(A, \Gamma^{-1}G\Gamma'^{-1}, \Lambda'^{-1}H\Lambda^{-1}).
\end{aligned}$$

□

6.3 The LRT when both Σ and Ψ are known

For a sample $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_n$ from a matrix variate normal population $N_{p,q}(M, \Sigma, \Psi)$, we consider testing a hypothesis H_0 against a one-sided alternative H_1 defined by

$$H_0 : M \in \mathcal{L}(V,W), \quad H_1 : M \in \mathcal{C}(V,W) \setminus \mathcal{L}(V,W). \tag{6.9}$$

In this section, we briefly summarize the likelihood ratio test (LRT) when both Σ and Ψ are known. Noting that

$$\sum_{i=1}^{n} ||\mathbb{X}_i - M||^2_{\Sigma,\Psi} = \sum_{i=1}^{n} ||\mathbb{X}_i - \overline{\mathbb{X}}||^2_{\Sigma,\Psi} + n||\overline{\mathbb{X}} - M||^2_{\Sigma,\Psi}$$

for $\overline{\mathbb{X}} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \mathbb{X}_i$, we derive the LRT statistic from (6.3) and Lemma 6.2.8 as follows.

$$
\begin{aligned}
&2\log\frac{\sup_{H_1}\prod_{i=1}^{n} f(\mathbb{X}_i, M, \Sigma, \Psi)}{\sup_{H_0}\prod_{i=1}^{n} f(\mathbb{X}_i, M, \Sigma, \Psi)} \\
&= n||\overline{\mathbb{X}} - \Pi_{\Sigma,\Psi}(\overline{\mathbb{X}}|\mathcal{L}(V, W))||^2_{\Sigma,\Psi} - n||\overline{\mathbb{X}} - \Pi_{\Sigma,\Psi}(\overline{\mathbb{X}}|\mathcal{C}(V, W))||^2_{\Sigma,\Psi} \\
&= n||\Pi_{\Sigma,\Psi}(\overline{\mathbb{X}}|\mathcal{C}(V, W)) - \Pi_{\Sigma,\Psi}(\overline{\mathbb{X}}|\mathcal{L}(V, W))||^2_{\Sigma,\Psi} \\
&= nT_{V,W}(\overline{\mathbb{X}}, \Sigma, \Psi). \qquad (6.10)
\end{aligned}
$$

It is well known that the null distribution of the LRT statistic (6.10) is a $\overline{\chi^2}$ distribution, which is a weighted sum of χ^2 distributions with different degrees of freedom. See Robertson et al. [7] and Silvapulle and Sen [11] for details of the $\overline{\chi^2}$ tests. The weights depend on the geometrical property of the alternative cone $\mathcal{C}(V, W)$. It is important but difficult to evaluate the weight of a $\overline{\chi^2}$ distribution. See also Shapiro [10] and Miwa et al. [5] for extended arguments of derivation of weights of $\overline{\chi^2}$ distribution.

We denote by $P_{\mathbb{X}|M,\Sigma,\Psi}(E)$ the probability of an event E when $\mathbb{X}_i \sim N_{p,q}(M, \Sigma, \Psi)$ $(i = 1, 2, \ldots, n)$. The next theorem implies that the probability does not depend on M under the null hypothesis.

Theorem 6.3.1 *If $M \in \mathcal{L}(V, W)$, for any real c*

$$P_{\mathbb{X}|M,\Sigma,\Psi}(T_{V,W}(\overline{\mathbb{X}}, \Sigma, \Psi) \geq c) = P_{\mathbb{X}|O,\Sigma,\Psi}(T_{V,W}(\overline{\mathbb{X}}, \Sigma, \Psi) \geq c).$$

6.4 When Ψ Is Known but Σ Is Unknown

In this section, we consider testing the hypotheses (6.9) when Ψ is known but Σ is unknown. The problem is a matrix variate version of problems discussed in Sasabuchi et al. [9], Sasabuchi [8], and Hu and Banerjee [2]. Our purpose is to extend their results.

Let $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_n$ be a sample of size n from a matrix variate normal population $N_{p,q}(M, \Sigma, \Psi)$D. Under the assumption that Ψ is known and M has no constraints, the maximum likelihood estimators of M and Σ are as follows:

$$\overline{\mathbb{X}} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{X}_i, \qquad (6.11)$$

$$S = \frac{1}{nq}\sum_{i=1}^{n}(\mathbb{X}_i - \overline{\mathbb{X}})\Psi^{-1}(\mathbb{X}_i - \overline{\mathbb{X}})', \qquad (6.12)$$

and $\overline{\mathbb{X}}$ and S are independently distributed as

$$\overline{\mathbb{X}} \sim N_{p,q}(M, \Sigma, \frac{1}{n}\Psi), S \sim W(q(n-1), \Sigma).$$

See Kollo and von Rosen [3] and Srivastava et al. [12] for statistical inferences on matrix variate normal distributions.

For the hypotheses (6.9), we consider the following test statistic constructed by substituting Σ in (6.10) by its estimator S.

$$T_{V,W}(\overline{\mathbb{X}}, S, \Psi) = ||\Pi_{S,\Psi}(\overline{\mathbb{X}}|\mathcal{C}(V, W)) - \Pi_{S,\Psi}(\overline{\mathbb{X}}|\mathcal{L}(V, W))||^2_{S,\Psi}. \tag{6.13}$$

The statistics $T_{V,W}(\overline{\mathbb{X}}, S, \Psi)$ is a generalization of that studied by Sasabuchi et al. [9] but not the LRT statistics for (6.9).

In order to determine critical values or p values for the test statistic, we consider the probability

$$P_{\mathbb{X}|M,\Sigma,\Psi}(T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c) \tag{6.14}$$

for arbitrary c under H_0. Since $\overline{\mathbb{X}}$ and S are independent, the probability (6.14) does not depend on $M \in \mathcal{L}(V, W)$ when Σ is fixed.

To evaluate the probability (6.14) when Σ varies, we consider a linear transformation $\mathbb{X}^*_i = \Gamma\mathbb{X}_i$. Then, $\mathbb{X}_i \sim N_{p,q}(M, \Sigma, \Psi)$ is equivalent to

$$\mathbb{X}^*_i \sim N_{p,q}(M^*, \Sigma^*, \Psi) \quad \text{where } M^* = \Gamma M,\ \Sigma^* = \Gamma\Sigma\Gamma'. \tag{6.15}$$

if $\Gamma \in \mathbb{R}^{p,p}$ is nonsingular. The hypotheses H_0 and H_1 are rewritten as

$$H^*_0 : M^* \in \mathcal{L}(V\Gamma^{-1}, W) \text{ and } H^*_1 : M^* \in \mathcal{C}(V\Gamma^{-1}, W) \setminus \mathcal{L}(V\Gamma^{-1}, W) \tag{6.16}$$

by Lemma 6.2.4. The maximum likelihood estimators of M^* and Σ^* are given by

$$\overline{\mathbb{X}}^* = \frac{1}{n}\sum_{i=1}^{n}\mathbb{X}^*_i = \Gamma\overline{\mathbb{X}},$$
$$S^* = \frac{1}{nq}\sum_{i=1}^{n}(\mathbb{X}^*_i - \overline{\mathbb{X}}^*)\Psi^{-1}(\mathbb{X}^*_i - \overline{\mathbb{X}}^*)' = \Gamma S\Gamma',$$

respectively, and we consider a test statistic defined by

$$T_{V\Gamma^{-1},W}(\mathbb{X}^*, S^*, \Psi) = ||\Pi_{S^*,\Psi}(\overline{\mathbb{X}}^*|\mathcal{C}(V\Gamma^{-1}, W)) - \Pi_{S^*,\Psi}(\overline{\mathbb{X}}^*|\mathcal{L}(V, W))||^2_{S^*,\Psi}$$

The following invariant property of the test statistic is shown by Lemmas 6.2.1, 6.2.2 and the Eq. (6.2).

Lemma 6.4.1 *When* $\mathbb{X}^*_i = \Gamma\mathbb{X}_i$, $T_{V\Gamma^{-1},W}(\mathbb{X}^*, S^*, \Psi) = T_{V,W}(\mathbb{X}, S, \Psi)$.

Lemma 6.4.2 *If* $\mathcal{L}(V\Gamma, W) = \mathcal{L}(V, W)$ *and* $\mathcal{C}(V, W) \subset \mathcal{C}(V\Gamma, W)$,

$$P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} \leq P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V\Gamma,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}$$

for any c.

Theorem 6.4.1 *Suppose that $\Gamma \in \mathbb{R}^{p,p}$ is a nonsingular matrix satisfying $\Sigma_1 = \Gamma\Sigma_2\Gamma'$ for Σ_1, Σ_2. If $\mathcal{L}(V\Gamma, W) = \mathcal{L}(V, W)$ and $\mathcal{C}(V, W) \subset \mathcal{C}(V\Gamma, W)$,*

$$P_{\mathbb{X}|M,\Sigma_1,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} \geq P_{\mathbb{X}|M,\Sigma_2,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}.$$

Proof For $\mathbb{X}_i \sim N_{p,q}(M, \Sigma_1, \Psi)$, we put $\mathbb{X}_i^* = \Gamma^{-1}\mathbb{X}_i$. Then, $\mathbb{X}_i^* \sim N_{p,q}(M^*, \Sigma_2, \Psi)$, and from Lemmas 6.4.1 and 6.4.2 we have

$$\begin{aligned} P_{\mathbb{X}|M,\Sigma_1,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} &= P_{\mathbb{X}^*|M^*,\Sigma_2,\Psi}\{T_{V\Gamma,W}(\overline{\mathbb{X}^*}, S^*, \Psi) \geq c\} \\ &\geq P_{\mathbb{X}^*|M^*,\Sigma_2,\Psi}\{T_{V,W}(\overline{\mathbb{X}^*}, S^*, \Psi) \geq c\}. \end{aligned}$$ □

Roughly stating, to evaluate the supremum of $P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) > c\}$ when Σ varies, it is sufficient to find maximal sets of $\mathcal{C}(V\Gamma, W)$ when Γ varies. We shall recall Theorem 6.2.1.

Let $\boldsymbol{u}$ be a nonnegative combination of $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_s$ (i.e., $\boldsymbol{u} \in \mathcal{C}\langle V\rangle$). We consider another test statistic defined by

$$\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) = ||\Pi_{S,\Psi}(\overline{\mathbb{X}}|\mathcal{C}(\boldsymbol{u}', W)) - \Pi_{S,\Psi}(\overline{\mathbb{X}}|\mathcal{L}(V, W))||^2_{S,\Psi}. \tag{6.17}$$

Note that $\mathcal{L}(V, W) \subset \mathcal{C}(\boldsymbol{u}', W)$.

Theorem 6.4.2 *For any positive definite $\Sigma \in \mathbb{R}^{p,p}$, we put*

$$\Sigma_n = \Xi_n \Sigma \Xi_n' \quad (n = 1, 2, \ldots), \tag{6.18}$$

where Ξ_n is defined in (6.4). Then, if $\boldsymbol{u}'\boldsymbol{v}_i > 0$ for all $i = 1, 2, \ldots, s$,
(i) $P_{\mathbb{X}|O,\Sigma_n,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}$ is increasing in n, and
(ii) $\lim_{n\to\infty} P_{\mathbb{X}|O,\Sigma_n,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} = P_{\mathbb{X}|O,\Sigma,\Psi}\{\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}$.

Proof From Theorem 6.2.1, for Ξ_n $(n = 1, 2, \ldots)$, we have

$$\mathcal{C}(V\Xi_1, W) \subset \mathcal{C}(V\Xi_2, W) \subset \cdots \subset \mathcal{C}(V\Xi_n, W) \subset \cdots$$

and $\mathcal{L}(V\Xi_n, W) = \mathcal{L}(V, W)$ for all $n = 1, 2, \ldots$. Therefore, the probability $P_{\mathbb{X}|O,\Sigma,\Psi}\{T_{V\Xi_n,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}$ is increasing in n by Theorem 6.4.1. Putting $\mathbb{X}_i^* = \Xi_n\mathbb{X}_i$, since $\mathbb{X}_i^* \sim N_{p,q}(O, \Sigma_n, \Psi)$, we have

$$\begin{aligned} P_{\mathbb{X}|O,\Sigma,\Psi}\{T_{V\Xi_n,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} &= P_{\mathbb{X}^*|O,\Sigma_n,\Psi}\{T_{V,W}(\overline{\mathbb{X}^*}, S^*, \Psi) \geq c\} \\ &= P_{\mathbb{X}|O,\Sigma_n,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}. \end{aligned}$$

by Lemma 6.4.1. Moreover, Theorem 6.2.1 implies

$$\lim_{n\to\infty} T_{V\Xi_n,W}(\overline{\mathbb{X}}, S, \Psi) = \tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi),$$

which completes the proof. □

Before we prove the similarity of the test statistic $\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi)$, we give a simple lemma.

Lemma 6.4.3 *When $\boldsymbol{u} \in \mathcal{L}\langle V \rangle$, $\Gamma'\boldsymbol{u}$ is orthogonal to the subspace $\mathcal{L}(V\Gamma, \bullet) \subset \mathbb{R}^p$ for any nonsingular $\Gamma \in \mathbb{R}^{p,p}$.*

Proof When $\boldsymbol{x} \in \mathcal{L}(V\Gamma, \bullet)$, it holds that $\boldsymbol{v}_i'\Gamma\boldsymbol{x} = 0$ for all $i = 1, 2, \ldots, s$. Since $\boldsymbol{u} = \sum_{i=1}^{s} c_i \boldsymbol{v}_i$, we have $(\Gamma'\boldsymbol{u})'\boldsymbol{x} = \sum_{i=1}^{x} c_i \boldsymbol{v}_i'\Gamma\boldsymbol{x} = 0$. □

Theorem 6.4.3 *For any $\boldsymbol{u} \in \mathcal{L}\langle V \rangle$ and any positive definite $\Sigma \in \mathbb{R}^{p,p}$,*

$$P_{\mathbb{X}|O,\Sigma,\Psi}(\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c) = P_{\mathbb{X}|O,I_p,\Psi}(\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c).$$

Proof Suppose $\mathbb{X}_i \sim N_{p,q}(M, \Sigma, \Psi)$. Putting $\mathbb{X}_i^* = \Sigma^{-1/2}\mathbb{X}_i$, $\mathbb{X}_i^* \sim N_{p,q}(M^*, I_p, \Psi)$. Then, we have

$$P_{\mathbb{X}|O,\Sigma,\Psi}(\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c) = P_{\mathbb{X}^*|O,I_p,\Psi}(\tilde{T}_{\Sigma^{1/2}\boldsymbol{u},W}(\overline{\mathbb{X}^*}, S^*, \Psi) \geq c).$$

If there exists an orthogonal matrix $U \in \mathbb{R}^{p,p}$ such that

$$\mathcal{L}(V\Sigma^{1/2}U, W) = \mathcal{L}(V, W) \text{ and } \mathcal{C}(\boldsymbol{u}'\Sigma^{1/2}U, W) = \mathcal{C}(\boldsymbol{u}', W),$$

we have for $\mathbb{X}_i^{**} = U'\mathbb{X}_i^* \sim N_{p,q}(M^{**}, I_p, \Psi)$

$$P_{\mathbb{X}^*|O,I_p,\Psi}(\tilde{T}_{\Sigma^{1/2}\boldsymbol{u},W}(\overline{\mathbb{X}^*}, S^*, \Psi) \geq c) = P_{\mathbb{X}^{**}|O,I_p,\Psi}(\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}^{**}}, S^{**}, \Psi) \geq c),$$

and the proof will be completed. From Lemma 6.4.3, $\boldsymbol{u}$ and $\Sigma^{1/2}\boldsymbol{u}$ are orthogonal to $\mathcal{L}(V, \bullet)$ and to $\mathcal{L}(V\Sigma^{1/2}, \bullet)$, respectively. This implies that there exists an orthogonal matrix $U \in \mathbb{R}^{p,p}$ such that $U'\mathcal{L}(V\Sigma^{1/2}, \bullet) = \mathcal{L}(V, \bullet)$ and $U'\Sigma^{1/2}\boldsymbol{u} = k\boldsymbol{u}$ $(k > 0)$, that is, $\mathcal{L}(V\Sigma^{1/2}U, \bullet) = \mathcal{L}(V, \bullet)$ and $\mathcal{C}(\boldsymbol{u}'\Sigma^{1/2}U, \bullet) = \mathcal{C}(\boldsymbol{u}', \bullet)$. Therefore, from Lemma 6.2.5, we have $\mathcal{L}(V\Sigma^{1/2}U, W) = \mathcal{L}(V, W)$ and $\mathcal{C}(\boldsymbol{u}'\Sigma^{1/2}U, W) = \mathcal{C}(\boldsymbol{u}', W)$. This completes the proof. □

Corollary 6.4.1 *For any $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathcal{L}\langle V \rangle$ and any positive definite $\Sigma \in \mathbb{R}^{p,p}$,*

$$P_{\mathbb{X}|O,\Sigma,\Psi}(\tilde{T}_{\boldsymbol{u}_1,W}(\overline{\mathbb{X}}, S, \Psi) \geq c) = P_{\mathbb{X}|O,\Sigma,\Psi}(\tilde{T}_{\boldsymbol{u}_2,W}(\overline{\mathbb{X}}, S, \Psi) \geq c).$$

Proof By considering $\mathbb{X}_i^* = \Gamma\mathbb{X}_i$ where $\Gamma \in \mathbb{R}^{p,p}$ satisfies $\boldsymbol{u}_2 = \Gamma\boldsymbol{u}_1$, it is obvious from Theorem 6.4.3. □

The following is the main result in this chapter.

Theorem 6.4.4 *Suppose $\boldsymbol{u} \in \mathcal{C}\langle V \rangle$. Then*
(i) $P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} \leq P_{\mathbb{X}|M,\Sigma,\Psi}\{\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}$ *for any M and Σ,*
(ii) $\sup_{H_0} P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} = P_{\mathbb{X}|O,I,\Psi}\{\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}$.

Proof (i) Since $\mathcal{C}(V, W) \subset \mathcal{C}(\boldsymbol{u}', W)$ by Lemma 6.2.6, we accomplish the proof in the same manner as Lemma 6.4.2. (ii) Since $\mathcal{C}(V, W)$ has an interior point, there exists $\boldsymbol{u}^* \in \mathcal{L}\langle V \rangle$ satisfying $\boldsymbol{v}_i'\boldsymbol{u}^* > 0$ for $i = 1, 2, \ldots, s$. Then, from Theorem 6.4.2 and (i), we have

$$\sup_{H_0} P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} = P_{\mathbb{X}|O,I,\Psi}\{\tilde{T}_{\boldsymbol{u}^*,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}.$$

From Corollary 6.4.1, we have

$$\sup_{H_0} P_{\mathbb{X}|M,\Sigma,\Psi}\{T_{V,W}(\overline{\mathbb{X}}, S, \Psi) \geq c\} = P_{\mathbb{X}|O,I,\Psi}\{\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi) \geq c\}. \qquad \square$$

Theorem 6.4.4 suggests a class of similar test statistics which are more powerful than $T_{V,W}(\overline{\mathbb{X}}, S, \Psi)$. Sasabuchi [8] and Hu and Banerjee [2] also give a class of similar and more powerful test statistic, but their class is more restrictive than our class.

Their similar test statistic corresponds to $\tilde{T}_{\boldsymbol{v}_i,W}(\overline{\mathbb{X}}, S, \Psi)$ (abbreviated by $\tilde{T}_i$ below) for $i = 1, 2, \ldots, s$ in our notation. As noted by Sasabuchi [8], the statistic $\tilde{T}_i$ is sensitive only for subrestriction $\mathcal{C}(\boldsymbol{v}_i', W)$, and we cannot choose an optimal one among $\{\tilde{T}_1, \tilde{T}_2, \ldots, \tilde{T}_s\}$. Sasabuchi [8] recommended to use an omnibus test statistic given by $T_{\min} = \min_{1 \leq i \leq s} \tilde{T}_i$ instead of $\tilde{T}_i$, although $T_{\min}$ is not similar and thus is inadmissible.

By Theorem 6.4.4, we can propose an alternative similar test statistic which is equally sensitive for all subrestrictions $\mathcal{C}(\boldsymbol{v}_1', W), \ldots, \mathcal{C}(\boldsymbol{v}_s', W)$. That is, we choose a vector $\boldsymbol{u}$ which indicates a center direction of $\mathcal{C}(V, \bullet)$, and construct the test statistics $\tilde{T}_{\boldsymbol{u},W}(\overline{\mathbb{X}}, S, \Psi)$. For example, if $V = I_p$ (i.e., $\mathcal{C}(V, \bullet)$ is a positive orthant), it may be reasonable to choose $\boldsymbol{u} = (1, 1, \ldots, 1)'$.

Acknowledgments The authors thank the referee for valuable comments. This work was supported by JSPS KAKENHI Grant Number 24540131.

References

1. Bartholomew, D. J. (1959). A test of homogeneity for ordered alternatives. *Biometrika*, *46*, 36–48.
2. Hu, X., & Banerjee, A. (2012). On the test for the homogeneity of a parameter matrix with some rows constrained by synchronized order restrictions. *Journal of Multivariate Analysis*, *107*, 64–70.
3. Kollo, T., & von Rosen, D. (2005). *Advanced multivariate statistics with matrices*. Dordrecht: Springer.
4. Kudo, A. (1963). A multivariate analogue of the one-sided test. *Biometrika*, *50*, 403–418.
5. Miwa, T., Hayter, A. J., & Kuriki, S. (2003). The evaluation of general non-centered orthant probabilities. *Journal of the Royal Statistical Society Series B*, *65*, 223–234.
6. Perlman, M. D. (1969). One-sided testing problems in multivariate analysis. *The Annals of Mathematical Statistics*, *40*, 549–567.

7. Robertson, T., Wright, F. T., & Dykstra, R. L. (1988). *Order restricted statistical inferences*. New York: Wiley.
8. Sasabuchi, S. (2007). More powerful tests for homogeneity of multivariate normal mean vectors under an order restriction. *Sankhya*, *69*, 700–716.
9. Sasabuchi, S., Tanaka, K., & Tsukamoto, T. (2003). Testing homogeneity of multivariate normal mean vectors under an order restriction when the covariance matrices are common but unknown. *The Annals of Statistics*, *31*, 1517–1536.
10. Shapiro, A. (1988). Towards a unified theory of inequality constrained testing in multivariate analysis. *International Statistical Review*, *56*, 49–62.
11. Silvapulle, M. J., & Sen, P. K. (2005). *Constrained statistical inference; inequality, order, and shape restrictions*. New York: Wiley.
12. Srivastava, M. S., von Rosen, T., & von Rosen, D. (2009). Estimation and testing in general multivariate liner models with Kronecker product covariance structure. *Sankhya*, *71A*, 137–163.