Contents

1 A Taxonomy of Emerging Multilinear Discriminant Analysis Solutions for Biometric Signal Recognition **1**

Haiping Lu, K. N. Plataniotis and A. N. Venetsanopoulos

1.1 Introduction 1
1.2 Multilinear basics 4
1.3 Multilinear discriminant analysis 7
1.4 Empirical Comparison of MLDA variants on Face Recognition 15
1.5 Conclusions 21
Appendix: Multilinear decompositions 21

References **23**

1 A Taxonomy of Emerging Multilinear Discriminant Analysis Solutions for Biometric Signal Recognition

Haiping Lu[1], Konstantinos N. Plataniotis[2] and Anastasios N. Venetsanopoulos[3]

[1,2]The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, M5S 3G4, Canada
[3]Ryerson University, Toronto, Ontario, M5B 2K3, Canada
{haiping[1], kostas[2],anv[3]}@comm.toronto.edu

ABSTRACT

Biometric signals are mostly multidimensional objects, known as tensors. Recently, there has been a growing interest in multilinear discriminant analysis (MLDA) solutions operating directly on these tensorial data. However, the relationships among these algorithms and their connections to linear (vector-based) algorithms are not clear, and in-depth understanding is needed for further developments and applications. In this chapter, we introduce the basics needed in understanding existing MLDA solutions and then categorize them according to the multilinear projection employed, while pointing out their connections with traditional linear solutions at the same time. A number of commonly used objective criteria and initialization methods are discussed. Experiments are carried out on two public face databases to evaluate the performance of the MLDA variants, and the results show that MLDA (and multilinear learning algorithms in general) is a promising field with great research potential.

1.1 INTRODUCTION

Many biometric signals, such as fingerprint, palmprint, ear, face images and gait silhouettes sequences, are naturally multi-dimensional objects, which are formally referred to as tensor objects. The elements of a tensor are to be addressed by a number of indices [1]. The number of indices used in the description defines the order of the tensor object and each index defines one “mode”.

Gray-level biometric images[1], such as face images, are naturally second-order tensors with the column and row modes [2, 3]. Color biometric images are naturally third-order tensors with the column, row and color modes [4, 5]. Three-dimensional gray-level faces are naturally third-order tensors with the column, row and depth modes [6, 7], and the popular Gabor faces [8] are third-order tensors with the column, row and Gabor modes. In many surveillance applications, the (sequential) biometric signals observed in surveillance video sequences [9] are naturally higher-order tensors. Binary gait silhouette sequences, the input to most if not all gait recognition algorithms [10–12], as well as other gray-level biometric video sequences can be viewed as third-order tensors with the column, row and time modes. Naturally, color biometric video sequences are fourth-order tensors with the addition of a color mode.

For illustration, Figure 1.1 shows the natural representations of three commonly used biometric signals, a second-order face tensor with the column and row modes in Fig. 1.1(a), a third-order Gabor face [2, 8, 13] tensor with the column, row and Gabor modes in Fig. 1.1(b), and a third-order gait silhouette sequence tensor [14] with the column, row and time modes in Fig. 1.1(c).

The tensor space where a typical biometric tensor object is specified is often high-dimensional, and recognition methods operating directly on this space suffer from the so-called curse of dimensionality [15]. On the other hand, the classes of a particular biometric signal, such as face images, are usually highly constrained and belong to a subspace, a manifold of intrinsically low dimension [15, 16]. Feature extraction or dimensionality reduction is thus an attempt to transform a high-dimensional data set into a low-dimensional space of equivalent representation while retaining most of the underlying structure [17]. Traditionally, feature extraction algorithms operate on one-dimensional objects, i.e., first-order tensors (vectors), and any tensor object with order greater than one, such as images and videos, have to be reshaped (vectorized) into vectors first before processing. However, it is well understood that reshaping breaks the natural structure and correlation in the original data, removing redundancies and/or higher order dependencies present in the original data set and losing potentially more compact or useful representations that can be obtained in the original form.

By recognizing the fact that tensor objects are naturally multi-dimensional objects instead of one-dimensional objects, multilinear feature extraction algorithms [2, 14, 18–20] operating directly on the tensorial representations rather than their vectorized versions are emerging, partly due to the recent development in multilinear algebra [1, 21, 22]. The multilinear principal component analysis (MPCA) framework [14] [2] attempts to determine a multilinear projection that projects the original tensor objects into a lower-dimensional tensor subspace while preserving the variation in the original data. It can be further extended through the combination with classical approaches [14, 18, 25] and has achieved good results when applied to the gait recognition problem. Nonetheless, MPCA is an unsupervised method and the class

[1] The discussion here applies to images in general, however, since this book is on biometrics, we put emphasis on biometric signals in this chapter.

[2] An earlier version with a slightly different approach appears in [23] and a different formulation is in [24].

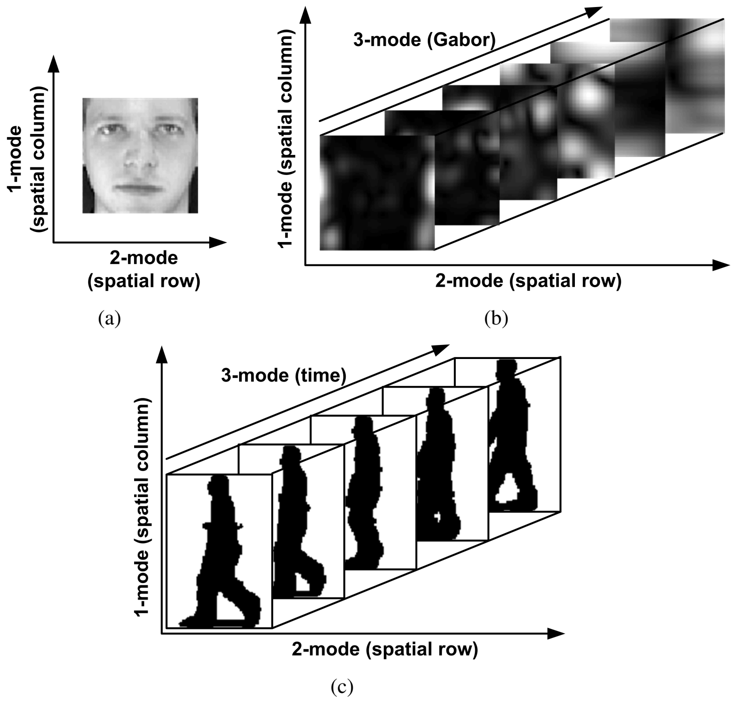

Fig. 1.1 Biometric data represented naturally as tensors: (a) a 2-D face tensor, (b) a 3-D Gabor-face tensor, (c) a 3-D gait (silhouette) tensor.

information is not used in the feature extraction process. There has been a growing interest in the development of supervised multilinear feature extraction algorithms. A two-dimensional linear discriminant analysis (2DLDA) was proposed in [26], and later a more general extension, the Discriminant Analysis with Tensor Representation (DATER)[3] was proposed in [2]. They maximize a tensor-based scatter ratio criterion and the application to the face recognition problem showed better recognition results than linear discriminant analysis (LDA). In [19], a so-called general tensor discriminant analysis (GTDA) algorithm is proposed by maximizing a scatter difference criterion and it is used as a preprocessing step in tensorial gait data classification [19]. All these methodologies are based on the tensor-to-tensor projection (TTP). The so-called Tensor Rank-one Discriminant Analysis (TR1DA) algorithm [27, 28], which uses the scatter difference criterion, obtains a number of rank-one projections from the repeatedly-calculated residues of the original tensor data and it can be viewed as a tensor-to-vector projection (TVP). This "greedy" approach is a heuristic method originally proposed in [29] for tensor approximation. In [30],

[3]Here, we adopt the name that was used when the algorithm was first proposed, which is more commonly refereed to in the literature.

an uncorrelated multilinear discriminant analysis (UMLDA) approach is proposed to extract uncorrelated features through TVP. The extensions of linear graph-embedding algorithms were also introduced similarly in [31–35].

In this chapter, we focus primarily on the development of supervised multilinear methodologies, in particular the multilinear discriminant analysis (MLDA) algorithms, the multilinear extensions of the well-known LDA algorithm. The objective is to answer the following two questions regarding MLDA so that the interested researchers/practitioners can grasp multilinear concepts with ease and clarity for practical usage and further research/development:

1. What are the various multilinear projections and how are they related to traditional linear projection?

2. What are the relationships (similarities and differences) among the existing MLDA variants?

First in Section 1.2, basic multilinear algebra is reviewed and the commonly used tensor distance measure is shown to be equivalent to the Euclidean distance for vectors. Next, Section 1.3 discusses various multilinear projections including linear projection: from vector to vector, from tensor to tensor and from tensor to vector, based on which the two general categories of MLDA are introduced. Commonly used separation criteria and initialization methods are then discussed and the underlying connections between the LDA and the MLDA variants are revealed. Subsequently, a taxonomy of the existing MLDA variants is suggested. Finally, empirical studies are presented in Section 1.4 and conclusions are drawn in Section 1.5.

1.2 MULTILINEAR BASICS

Before discussions on the multilinear discriminant analysis solutions for biometric signals, it is necessary to review some basic multilinear algebra, including the notations and some basic multilinear operations. To pursue further in this topic, [1,21,22,29,36] are excellent references. In addition, the equivalent vector interpretation of a commonly used tensor distance measure is derived.

1.2.1 Notations

The notations in this chapter follow the conventions in the multilinear algebra, pattern recognition and adaptive learning literature. Vectors are denoted by lowercase boldface letters, e.g., $\mathbf{x}$; matrices by uppercase boldface, e.g., $\mathbf{U}$; and tensors by calligraphic letters, e.g., $\mathcal{A}$. Their elements are denoted with indices in brackets. Indices are denoted by lowercase letters and span the range from 1 to the uppercase letter of the index, e.g., $n = 1, 2, ..., N$. Throughout this chapter, the discussion is restricted to real-valued vectors, matrices and tensors since the biometric applications that we are interested in involve real data only, such as gray-level/color face images and binary gait silhouette sequences.

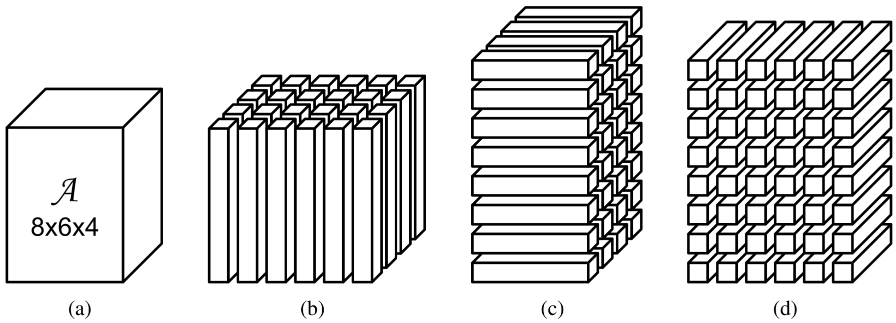

Fig. 1.2 Illustration of the n-mode vectors: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8\times 6\times 4}$, (b) the 1-mode vectors, (c) the 2-mode vectors, and (d) the 3-mode vectors.

1.2.2 Basic multilinear algebra

An Nth-order tensor is denoted as: $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$. It is addressed by N indices i_n, $n = 1, ..., N$, and each i_n addresses the n-mode of $\mathcal{A}$. The n-mode product of a tensor $\mathcal{A}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n\times I_n}$, denoted by $\mathcal{A} \times_n \mathbf{U}$, is a tensor with entries:

$$(\mathcal{A} \times_n \mathbf{U})(i_1, ..., i_{n-1}, j_n, i_{n+1}, ..., i_N) = \sum_{i_n} \mathcal{A}(i_1, ..., i_N) \cdot \mathbf{U}(j_n, i_n). \quad (1.1)$$

The scalar product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$ is defined as:

$$< \mathcal{A}, \mathcal{B} >= \sum_{i_1}\sum_{i_2} ... \sum_{i_N} \mathcal{A}(i_1, i_2, ..., i_N) \cdot \mathcal{B}(i_1, i_2, ..., i_N) \quad (1.2)$$

and the Frobenius norm of $\mathcal{A}$ is defined as $\parallel \mathcal{A} \parallel_F = \sqrt{< \mathcal{A}, \mathcal{A} >}$. The "$n$-mode vectors" of $\mathcal{A}$ are defined as the I_n-dimensional vectors obtained from $\mathcal{A}$ by varying the index i_n while keeping all the other indices fixed. A rank-1 tensor $\mathcal{A}$ equals to the outer product of N vectors: $\mathcal{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, which means that

$$\mathcal{A}(i_1, i_2, ..., i_N) = \mathbf{u}^{(1)}(i_1) \cdot \mathbf{u}^{(2)}(i_2) \cdot ... \cdot \mathbf{u}^{(N)}(i_N) \quad (1.3)$$

for all values of indices. Unfolding $\mathcal{A}$ along the n-mode is denoted as $\mathbf{A}_{(n)} \in \mathbb{R}^{I_n\times(I_1\times ...\times I_{n-1}\times I_{n+1}\times ...\times I_N)}$, and the column vectors of $\mathbf{A}_{(n)}$ are the n-mode vectors of $\mathcal{A}$.

Figures 1.2(b), 1.2(c) and 1.2(d) give visual illustrations of the 1-mode, 2-mode and 3-mode vectors of the third-order tensor $\mathcal{A}$ in Fig. 1.2(a), respectively. Figure 1.3(a) shows the 1-mode unfolding of the tensor $\mathcal{A}$ in Fig. 1.2(a) and Fig. 1.3(b) demonstrates how the 1-mode multiplication $\mathcal{A} \times_1 \mathbf{B}$ is obtained. The product $\mathcal{A} \times_1 \mathbf{B}$ is computed as the inner product between the 1-mode vector of $\mathcal{A}$ and the rows of $\mathbf{B}$. In the 1-mode multiplication, each 1-mode vector of $\mathcal{A}$ ($\in \mathbb{R}^8$) is projected by $\mathbf{B} \in \mathbb{R}^{3\times 8}$ to obtain a vector ($\in \mathbb{R}^3$), as the differently shaded vectors indicate in Fig. 1.3(b).

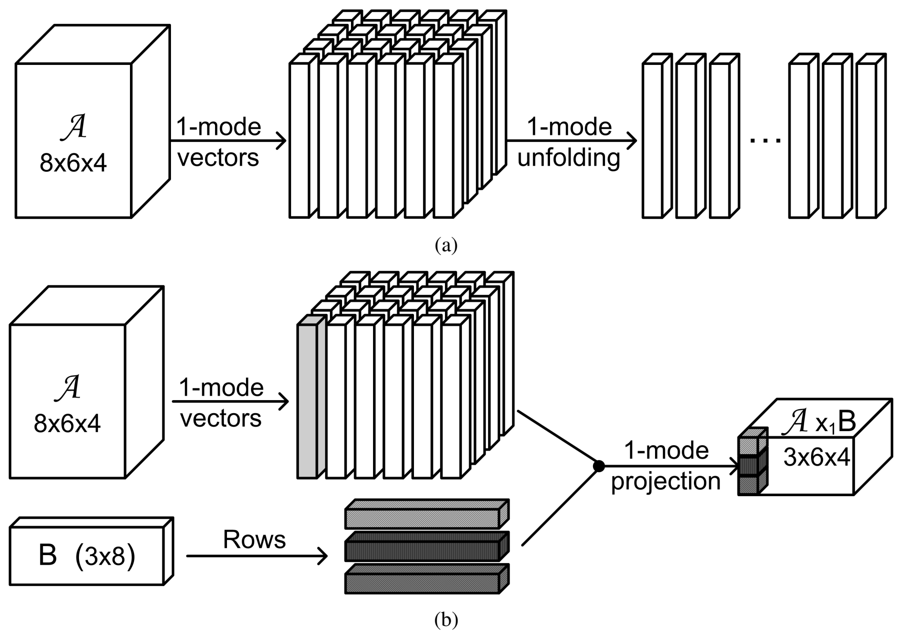

Fig. 1.3 Visual illustration of (a) the n-mode (1-mode) unfolding and (b) the n-mode (1-mode) multiplication.

1.2.3 Tensor distance measure

To measure the distance between tensors $\mathcal{A}$ and $\mathcal{B}$, the Frobenius norm is used in [2]: $dist(\mathcal{A}, \mathcal{B}) = \parallel \mathcal{A} - \mathcal{B} \parallel_F$. Let $vec(\mathcal{A})$ be the vector representation (vectorization) of $\mathcal{A}$, then it is straightforward to show that

Proposition 1 $dist(\mathcal{A}, \mathcal{B}) = \parallel vec(\mathcal{A}) - vec(\mathcal{B}) \parallel_2$

I.e.,the Frobenius norm of the difference between two tensors equals to the Euclidean distance of their vectorized representations, since the Frobenius norm is a point-based measurement as well [37] and it does not take the structure of a tensor into account.

1.3 MULTILINEAR DISCRIMINANT ANALYSIS

The linear discriminant analysis (LDA) [38] is a classical algorithm that has been successfully applied and extended to various biometric signal recognition problems [15, 39–42]. The recent advancement in multilinear algebra [1, 21] led to a number of multilinear extensions of the LDA, multilinear discriminant analysis (MLDA), being proposed for the recognition of biometric signals using their natural tensorial representation [2, 19, 28, 30].

In general, MLDA seeks a multilinear projection that maps the input data from one space to another (lower-dimensional, more discriminative) space. Therefore, we need to understand what is a multilinear projection before proceeding to the MLDA solutions. In this section, we first propose a categorization of the various multilinear projections in terms of the input and output of the projection: the traditional vector-to-vector projection (VVP), the tensor-to-tensor projection (TTP) and the tensor-to-vector (TVP) projection[4]. Based on the categorization of multilinear projections, we discuss two general formulations of MLDA: the MLDA based on the tensor-to-tensor projection (MLDA-TTP) and the MLDA based on the tensor-to-vector projection (MLDA-TVP). Commonly used separation criteria and initialization methods are then presented. Furthermore, the relationships between the LDA, MLDA-TTP and MLDA-TVP are investigated and a taxonomy of the existing MLDA variants is suggested.

1.3.1 Vector-to-Vector projection (VVP)

Linear projection is a standard transform used widely in various applications [38, 43]. A linear projection takes a vector $\mathbf{x} \in \mathbb{R}^I$ and projects it to $\mathbf{y} \in \mathbb{R}^P$ using a projection matrix $\mathbf{U} \in \mathbb{R}^{I \times P}$:

$$\mathbf{y} = \mathbf{U}^T \mathbf{x}. \tag{1.4}$$

In typical pattern recognition applications, $P << I$. Therefore, linear projection is a vector-to-vector projection (VVP) and it requires the vectorization of an input before projection. Figure 1.4(a) illustrates the VVP of a tensor object $\mathcal{A}$. The classical LDA algorithm employs VVP.

1.3.2 Tensor-to-Tensor projection (TTP)

Besides the traditional VVP, we can also project a tensor to another tensor (of the same order), which is named as tensor-to-tensor projection (TTP) in this chapter. An Nth-order tensor $\mathcal{X}$ resides in the tensor (multilinear) space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$, where $\bigotimes$ denotes the Kronecker product [43]. Thus the tensor (multilinear) space

[4] Multilinear projections are closely related to multilinear/tensor decompositions, which are included in the appendix for completeness. They share some mathematical similarities but they are from different perspectives.

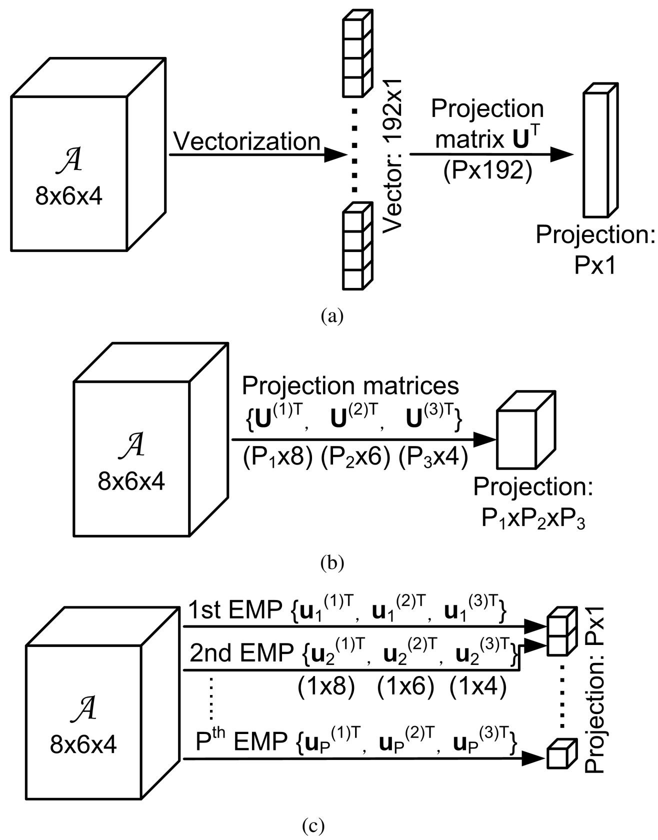

Fig. 1.4 Illustration of (a) vector-to-vector projection (VVP), (b) tensor-to-tensor projection (TTP), (c) tensor-to-vector projection (TVP).

can be viewed as the Kronecker product of N vector (linear) spaces $\mathbb{R}^{I_1}$, $\mathbb{R}^{I_2}$, ..., $\mathbb{R}^{I_N}$. For the projection of a tensor $\mathcal{X}$ in a tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$ to another tensor $\mathcal{Y}$ in a lower-dimensional tensor space $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} ... \bigotimes \mathbb{R}^{P_N}$, where $P_n < I_n$ for all n, N projection matrices $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, ..., N\}$ are used so that

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}. \tag{1.5}$$

Figure 1.4(b) demonstrates the TTP of a tensor object $\mathcal{A}$ to a smaller tensor of size $P_1 \times P_2 \times P_3$. How this multilinear projection is carried out can be understood better by referring to the illustration on the n-mode multiplication in Fig. 1.3(b). Many multilinear algorithms [2, 14, 19] have been developed through solving such a TTP.

1.3.3 Tensor-to-Vector projection (TVP)

In our recent work [30], we introduced a multilinear projection from a tensor space to a vector space, called the tensor-to-vector projection (TVP). The projection from a tensor to a scalar is considered first. A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ is projected to a point y as:

$$y = \mathcal{X} \times_1 \mathbf{u}^{(1)^T} \times_2 \mathbf{u}^{(2)^T} ... \times_N \mathbf{u}^{(N)^T}, \tag{1.6}$$

which can also be written as the following inner product:

$$y =< \mathcal{X}, \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)} > . \tag{1.7}$$

Let $\mathcal{U} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, then we have $y =< \mathcal{X},\mathcal{U} >$. Such a multilinear projection $\{\mathbf{u}^{(1)^T}, \mathbf{u}^{(2)^T}, ..., \mathbf{u}^{(N)^T}\}$, named an elementary multilinear projection (EMP), is the projection of a tensor on a single multilinear projection direction, and it consists of one projection vector in each mode.

The projection of a tensor object $\mathcal{X}$ to $\mathbf{y} \in \mathbb{R}^P$ in a P-dimensional vector space consists of P EMPs

$$\{\mathbf{u}_p^{(1)^T}, \mathbf{u}_p^{(2)^T}, ..., \mathbf{u}_p^{(N)^T}\}, p = 1, ..., P, \tag{1.8}$$

which can be written compactly as $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^P$. Thus, this TVP is written as

$$\mathbf{y} = \mathcal{X} \times_{n=1}^N \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^P, \tag{1.9}$$

where the pth component of $\mathbf{y}$ is obtained from the pth EMP as:

$$\mathbf{y}(p) = \mathcal{X} \times_1 \mathbf{u}_p^{(1)^T} \times_2 \mathbf{u}_p^{(2)^T} ... \times_N \mathbf{u}_p^{(N)^T}. \tag{1.10}$$

Figure 1.4(c) shows the TVP of a tensor object $\mathcal{A}$ to a vector of size $P \times 1$. A number of recent multilinear algorithms [27, 28, 30, 35] [5] have been proposed with the objective of solving such a TVP.

1.3.4 MLDA-TTP

The multilinear extension of the LDA using the TTP is named MLDA-TTP hereafter. To formulate MLDA-TTP, the following definitions are introduced first.

Definition 1 *Let $\{\mathcal{A}_m, m = 1, ..., M\}$ be a set of M tensor samples in $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$. The between-class scatter of these tensors is defined as:*

$$\Psi_{B_{\mathcal{A}}} = \sum_{c=1}^{C} N_c \parallel \bar{\mathcal{A}}_c - \bar{\mathcal{A}} \parallel_F^2, \tag{1.11}$$

[5] TVP is referred to as the Rank-one projections in some works [27, 28, 35].

and the within-class scatter of these tensors is defined as:

$$\Psi_{W_{\mathcal{A}}} = \sum_{m=1}^{M} \parallel \mathcal{A}_m - \bar{\mathcal{A}}_{c_m} \parallel_F^2, \tag{1.12}$$

where C is the number of classes, N_c is the number of samples for class c, c_m is the class label for the mth sample $\mathcal{A}_m$, the mean tensor $\bar{\mathcal{A}} = \frac{1}{M}\sum_m \mathcal{A}_m$ and the class mean tensor $\bar{\mathcal{A}}_c = \frac{1}{N_c}\sum_{m,c_m=c} \mathcal{A}_m$.

Next, the n-mode scatter matrices are defined accordingly.

Definition 2 *The n-mode between-class scatter matrix of these samples is defined as:*

$$\mathbf{S}_{B_{\mathcal{A}}}^{(n)} = \sum_{c=1}^{C} N_c \cdot \left(\bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)}\right)\left(\bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)}\right)^T, \tag{1.13}$$

and the n-mode within-class scatter matrix of these samples is defined as:

$$\mathbf{S}_{W_{\mathcal{A}}}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)}\right)\left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)}\right)^T, \tag{1.14}$$

where $\bar{\mathbf{A}}_{c(n)}$ is the n-mode unfolded matrix of $\bar{\mathcal{A}}_c$.

From the definitions above, the following properties are derived:

Property 1 *Since $trace(\mathbf{A}\mathbf{A}^T) = \parallel \mathbf{A} \parallel_F^2$ and $\parallel \mathcal{A} \parallel_F^2 = \parallel \mathbf{A}_{(n)} \parallel_F^2$, $trace\left(\mathbf{S}_{B_{\mathcal{A}}}^{(n)}\right) = \sum_{c=1}^{C} N_c \parallel \bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)} \parallel_F^2 = \Psi_{B_{\mathcal{A}}}$ and $trace\left(\mathbf{S}_{W_{\mathcal{A}}}^{(n)}\right) = \sum_{m=1}^{M} \parallel \mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)} \parallel_F^2 = \Psi_{W_{\mathcal{A}}}$, for all n.*

The formal definition of the problem to be solved in MLDA-TTP is then described below:

A set of M training tensor objects $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_M\}$ is available. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ assumes values in the tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$, where I_n is the n-mode dimension of the tensor. The objective of MLDA-TPP is to find a multilinear mapping $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, ..., N\}$ from the original tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$ into a tensor subspace $\mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} ... \bigotimes \mathbb{R}^{P_N}$ (with $P_n < I_n$, for $n = 1, ..., N$):

$$\mathcal{Y}_m = \mathcal{X}_m \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}, m = 1, ..., M, \tag{1.15}$$

based on the optimization of a certain separation criterion, such that an enhanced separability between different classes is achieved.

The MLDA-TTP objective is to determine the N projection matrices $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, ..., N\}$ that maximize some class separation criterion, which is often in terms of $\Psi_{B_{\mathcal{Y}}}$ and $\Psi_{W_{\mathcal{Y}}}$. By making use of Property 1, the problem can be converted to N sub-problems in terms of $\mathbf{S}_{B\mathcal{Y}}^{(n)}$ and $\mathbf{S}_{W\mathcal{Y}}^{(n)}$, which employs the commonly-used alternating projection principal [1, 2, 14]. The pseudo-code implementation of a general MLDA-TTP algorithm is shown in Fig. 1.5. In each iteration k, for mode n, the input tensor samples are projected using the current projection matrices in all modes except n to obtain a set of Nth-order tensor samples, whose n-mode unfolding matrices are used to obtain $\mathbf{S}_{B\mathcal{Y}}^{(n)}$ and $\mathbf{S}_{W\mathcal{Y}}^{(n)}$.

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$ with class labels $\mathbf{c} \in \mathbb{R}^M$, P_n for $n = 1, ..., N$.

Output: Low-dimensional representations $\{\mathcal{Y}_m \in \mathbb{R}^{P_1 \times P_2 \times ... \times P_N}, m = 1, ..., M\}$ of the input tensor samples maximizing a separation criterion.

Algorithm:

Step 1: Initialize $\mathbf{U}_0^{(n)}$ for $n = 1, ..., N$.

Step 2 (Local optimization):

- For $k = 1 : K$
  - For $n = 1 : N$
    - Calculate $\{\mathcal{Y}_m = \mathcal{X}_m \times_1 \mathbf{U}_k^{(1)^T} ... \times_{n-1} \mathbf{U}_k^{(n-1)^T} \times_{n+1} \mathbf{U}_{k-1}^{(n+1)^T} ... \times_N \mathbf{U}_{k-1}^{(N)^T}, m = 1, ..., M\}$.
    - Calculate $\mathbf{S}_{B\mathcal{Y}}^{(n)}$ and $\mathbf{S}_{W\mathcal{Y}}^{(n)}$.
    - Set the matrix $\mathbf{U}_k^{(n)}$ to optimize a separation criterion.
  - If $k > 2$ and $\mathbf{U}_k^{(n)}$ converges for all n, set $\mathbf{U}^{(n)} = \mathbf{U}_k^{(n)}$ and break.

Step 3 (Projection): The feature tensor after projection is obtained as $\{\mathcal{Y}_m = \mathcal{X}_m \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}, m = 1, ..., M\}$.

Fig. 1.5 The pseudo-code implementation of a general MLDA-TTP.

1.3.5 MLDA-TVP

The multilinear extension of the LDA using the TVP is named MLDA-TVP and the formal definition of the problem to be solved in MLDA-TVP is described below:

A set of M training tensor objects $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_M\}$ is available. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ assumes values in the tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$, where I_n is the n-mode dimension of the tensor. The objective of MLDA-TVP is to find a set of P EMPs $\{\mathbf{u}_p^{(n)} \in \mathbb{R}^{I_n \times 1}, n = 1, ..., N\}_{p=1}^{P}$ mapping from the original tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$ into a vector subspace $\mathbb{R}^P$ (with $P < \prod_{n=1}^{N} I_n$):

$$\mathbf{y}_m = \mathcal{X}_m \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}, m = 1, ..., M, \tag{1.16}$$

based on the optimization of a certain separation criteria, such that an enhanced separability between different classes is achieved.

The MLDA-TVP objective is to determine the P projection bases in each mode $\{\mathbf{u}_p^{(n)} \in \mathbb{R}^{I_n \times 1}, n = 1, ..., N, p = 1, ..., P\}$ that maximize a class separation criterion. In MLDA-TVP, since the projected space is a vector space, the definition of scatter matrices in classical LDA can be followed. For the samples projected by the pth EMP $\{y_{m_p}, m = 1, ..., M\}$, where y_{m_p} is the projection of the mth sample by the pth EMP, the between-class scatter matrix and the within-class scatter matrix are defined as

$$S_{B_p}^{\mathbf{y}} = \sum_{c=1}^{C} N_c (\bar{y}_{c_p} - \bar{y}_p)^2, \tag{1.17}$$

and

$$S_{W_p}^{\mathbf{y}} = \sum_{m=1}^{M} (y_{m_p} - \bar{y}_{c_{m_p}})^2, \tag{1.18}$$

respectively, where $\bar{y}_p = \frac{1}{M} \sum_m y_{m_p}$, $\bar{y}_{c_p} = \frac{1}{N_c} \sum_{m, c_m = c} y_{m_p}$. Figure 1.6 is the pseudo-code implementation of a general MLDA-TVP algorithm. To solve the problem, the alternating projection principal is again employed. In each iteration k, for mode n, the input tensor samples are projected using the current projection vectors in all modes except n to obtain a set of vector samples and the problem is then converted to a number of classical LDA problems.

1.3.6 Separation criteria and initialization methods

Both MLDA-TTP and MLDA-TVP need to specify a class separation criterion to be optimized. One commonly used separation criterion is the ratio of the between-class scatter $\Psi_{B_\mathcal{Y}}$ or $S_{B_p}^{\mathbf{y}}$ and the within-class scatter $\Psi_{W_\mathcal{Y}}$ or $S_{W_p}^{\mathbf{y}}$: $\left(\frac{\Psi_{B_\mathcal{Y}}}{\Psi_{W_\mathcal{Y}}}\right)$ for MLDA-TTP or $\left(\frac{S_{B_p}^{\mathbf{y}}}{S_{W_p}^{\mathbf{y}}}\right)$ for MLDA-TVP [39], hereafter named SRatio.

Another separation criterion is the (weighted) difference between the between-class scatter $\Psi_{B_\mathcal{Y}}$ or $S_{B_p}^{\mathbf{y}}$ and the within-class scatter $\Psi_{W_\mathcal{Y}}$ or $S_{W_p}^{\mathbf{y}}$: $(\Psi_{B_\mathcal{Y}} - \zeta \Psi_{W_\mathcal{Y}})$ for MLDA-TTP or $(S_{B_p}^{\mathbf{y}} - \zeta \cdot S_{W_p}^{\mathbf{y}})$ for MLDA-TVP [44], hereafter named SDiff, where ζ is a parameter tuning the weight between the between-class and within-class scatters.

Since MLDA algorithms rely on the alternating projection principal, they are generally iterative and there is a need in choosing an initialization method. Commonly used initialization methods for MLDA-TTP are: pseudo-identity matrices (truncated identity matrices) and random matrices. Commonly used initialization methods for

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$ with class labels $\mathbf{c} \in \mathbb{R}^M$, the projected feature dimension P.

Output: Low-dimensional representations $\{\mathbf{y}_m \in \mathbb{R}^P, m = 1, ..., M\}$ of the input tensor samples maximizing a separation criterion.

Algorithm:

Step 1 (Stepwise optimization):

For $p = 1 : P$

- For $n = 1, ..., N$, initialize $\mathbf{u}_p^{(n)} \in \mathbb{R}^{I_n}$.
- For $k = 1 : K$
  - For $n = 1 : N$
    - Calculate $\{\mathbf{y}_m = \mathcal{X}_{m_p} \times_1 \mathbf{u}_{p_k}^{(1)^T} ... \times_{n-1} \mathbf{u}_{p_k}^{(n-1)^T} \times_{n+1} \mathbf{u}_{p_{k-1}}^{(n+1)^T} ... \times_N \mathbf{u}_{p_{k-1}}^{(N)^T}, m = 1, ..., M\}$.
    - Calculate the between-class and the within-class scatter matrices by treating $\{\mathbf{y}_m\}$ as the input vector samples, as in classical LDA.
    - Compute the vector $\mathbf{u}_{p_k}^{(n)}$ that optimizes a separation criterion.
  - If k>2 and $\mathbf{u}_{p_k}^{(n)}$ converges for all n, set $\mathbf{u}_p^{(n)} = \mathbf{u}_{p_k}^{(n)}$ and break.

Step 2 (Projection): The feature vector after projection is obtained as $\{\mathbf{y}_m(p) = \mathcal{X}_m \times_1 \mathbf{u}_p^{(1)^T} ... \times_N \mathbf{u}_p^{(N)^T}, p = 1, ..., P, m = 1, ..., M\}$.

Fig. 1.6 The pseudo-code implementation of a general MLDA-TVP.

MLDA-TVP are: all ones and random vectors. There are also initialization methods based on projections obtained from the n-mode vectors of the input tensor samples [30, 45].

1.3.7 Relationships between the LDA, MLDA-TTP and MLDA-TVP

To study the relationships between the LDA, MLDA-TTP and MLDA-TVP, it is beneficial to investigate what are the relationships between VVP, TTP and TVP first. It is easy to verify that VVP is the special case of TTP and TVP with $N = 1$. On the other hand, each projected element in TTP can be viewed as the projection of an EMP formed by taking one column from each of the projection matrices and thus the projected tensor is obtained through $\prod_{n=1}^{N} I_n$ interdependent EMPs in effect, while in TVP, the P EMPs obtained sequentially are not interdependent generally.

Furthermore, recall that the projection using an EMP $\{\mathbf{u}^{(1)^T}, \mathbf{u}^{(2)^T}, ..., \mathbf{u}^{(N)^T}\}$ can be written as $y =< \mathcal{X}, \mathcal{U} >$, it is then straightforward to show

Proposition 2 $y =< \mathcal{X}, \mathcal{U} >=< vec(\mathcal{X}), vec(\mathcal{U}) >= [vec(\mathcal{U})]^T vec(\mathcal{X})$.

Thus, an EMP is equivalent to a linear projection of $vec(\mathcal{X})$, the vectorized representation of $\mathcal{X}$, on a vector $vec(\mathcal{U})$. Since $\mathcal{U} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, Proposition 2 indicates that the EMP is in effect a linear projection with constraint on the projection vector such that it is the vectorized representation of a rank-one tensor. Compared with a projection vector of size $I \times 1$ in VVP specified by I parameters ($I = \prod_{n=1}^{N} I_n$ for an Nth-order tensor), an EMP in TVP can be specified by $\sum_{n=1}^{N} I_n$ parameters. Hence, to projection a tensor of size $\prod_{n=1}^{N} I_n$ to a vector of size $P \times 1$, the TVP needs to estimate only $P \cdot \sum_{n=1}^{N} I_n$ parameters, while the VVP needs to estimate $P \cdot \prod_{n=1}^{N} I_n$ parameters. The implication in pattern recognition problem is that the TVP has fewer parameters to estimate while being more constrained on the solutions, and the VVP has less constraint on the solutions sought while having more parameters to estimate.

The connections between the MLDA algorithms and the LDA algorithm can be revealed through the relationships among VVP, TTP and TVP. From the analysis above, LDA is a special case of MLDA-TTP and MLDA-TVP when $N = 1$, with the scatter ratio as the separation criterion. On the other hand, the MLDA-TTP is looking for interdependent EMPs while the EMPs sought sequentially in the MLDA-TVP are not interdependent generally. Furthermore, for the same projected vector size, the MLDA-TVP has fewer parameters to estimate while the projection to be solved are more constrained, and LDA has more parameters to estimate while the projection is less constrained.

1.3.8 A taxonomy of MLDA variants

With the two general formulations of MLDA, a taxonomy of the existing MLDA variants is given in Table 1.1, followed by brief descriptions of the four MLDA variants listed in the view of this taxonomy.

Table 1.1 A taxonomy of MLDA variants.

MLDA variants	Projection type	Separation criterion	Reference
DATER	TTP	SRatio	[2]
GTDA	TTP	SDiff	[19]
TR1DA	TVP	SDiff	[27, 28]
UMLDA	TVP	SRatio	[30]

From the taxonomy suggested in Table 1.1, it can been seen that the Discriminant Analysis with Tensor Representation (DATER) algorithm [2] is a specific realization of the MLDA-TTP, with the objective of maximizing the scatter ratio and using the pseudo-identity matrices for initialization. The General Tensor Discriminant Analysis (GTDA) algorithm [46] is an MLDA-TTP variant maximizing the scatter difference, where in each step of the iteration and in each mode, the tuning parameter ζ is determined to be the maximum eigenvalue of a mode-wise scatter ratio (which means that a different weighting between the between-class and within-class scatter is used in each mode and each iteration). The initialization in GTDA is done by setting the initial projection matrix to be all ones. The Tensor Rank-one Discriminant Analysis (TR1DA) algorithm [27, 28] is an MLDA-TVP variant maximizing the scatter difference. In each iteration, TR1DA calculates the residues of all tensor samples using the obtained EMPs, which is a heuristic greedy approach used in tensor approximation problem [29], and the residues are used as the input tensor samples in the next iteration. The selection of ζ in TR1DA is not addressed in [27, 28] and random initialization is employed in this MLDA variant. The uncorrelated multilinear discriminant analysis (UMLDA) algorithm [30, 47] is an MLDA-TVP variant maximizing the scatter ratio, while pursuing uncorrelated features, and a regularization procedure with parameter η was introduced to increase the estimated within-class scatter, resulting in better generalization. The initialization method used in [30] is based on the n-mode vectors.

1.4 EMPIRICAL COMPARISON OF MLDA VARIANTS ON FACE RECOGNITION

In this section, empirical performance comparison of MLDA variants is carried out on 2-D face images (second-order tensors). For experiments on third-order tensors, please refer to [14,30] for results on gait silhouette sequences, and [2,19] for results on the Gabor face/gait images. Two public face databases with a large number of samples per subject available for testing were used. One is the PIE (Pose, Illumination, and Expression) database from CMU [48] and the other is the extended Yale face database B (YaleB) [49,50].

For the MLDA variants, all the face images are cropped and normalized to 32×32 pixels (represented as second-order tensors), with 256 gray levels per pixel [6]. A random subset with L (=5, 10, 20, 30) samples per subject was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given L, the results averaged over 20 random splits [7] are reported in this chapter. The nearest neighbor classifier with the Euclidean distance measure was employed in classification for simplicity. The MLDA-TTP variants (DATER and

[6] The 32×32 face data was obtained from http://www.cs.uiuc.edu/homes/dengcai2/Data/FaceData.html.
[7] The reason for randomly selecting the training set and repeating 20 times is to reduce the dependency of the performance on a particular set of training data.

GTDA) produce features in tensor representation, which cannot be handled directly by the selected classifier. Since from Section 1.2.3, the tensor distance measured by the Frobenius norm is equivalent to the Euclidean distance between vectorized representations, the tensor features from MLDA-TTP are rearranged to vectors for easy classification and comparison, which is described in detail in Section 1.4.1 below. Besides the four MLDA variants listed in Table 1.1, the Fisherface algorithm [39], which is a classical LDA approach, and the uncorrelated LDA (ULDA) algorithm [51] are included for comparison between LDA and MLDA. For LDA and ULDA, a 32×32 face image is represented as a 1024×1 vector for input.

In the experiments, the number of iterations for the MLDA variants was set to 10. For DATER, GTDA and TR1DA, up to 300 features were tested. For UMLDA, up to 100 features were tested. The maximum number of features tested for LDA and ULDA was $C - 1$, where C is the number of subjects (classes) in training. For the TR1DA algorithm, we tested several values of ζ for each L and the best one for each L was used: $\zeta = 2$ for $L = 5$, $\zeta = 0.8$ for $L = 10$, $\zeta = 0.6$ for $L = 20, 30$. For UMLDA, a fixed regularization parameter $\eta = 5 \times 10^4$ was empirically chosen and all initial projection vectors are set to all ones ($\mathbf{1}$) for simplicity.

(a)

(b)

Fig. 1.7 Sample face images of one subject from (a) the CMU PIE database and (b) the YaleB database.

1.4.1 Feature rearrangement for MLDA-TTP

The MLDA-TTP algorithms produce features in tensorial representation. For tensor distance calculation, the Frobenius norm is commonly used [2]. By Proposition 1, it is equivalent to calculate the Euclidean distance of their vectorized representation. Therefore, in this study, we rearrange the tensor features obtained by MLDA-TTP to vectors for easy comparison. The MLDA-TTP algorithms obtain the highest-dimension projection ($P_n = I_n$ for $n = 1, ..., N$) first and then the TTP is viewed as $\prod_{n=1}^{N} I_n$ EMPs. The discriminability of each such EMP is calculated on the training set and the EMPs are arranged in descending discriminability so that a feature vector is obtained, as in [14] for the MPCA algorithm. The MLDA-TVP algorithms produce feature vectors directly so there is no such rearrangement necessary.

1.4.2 Face recognition results on PIE database

The CMU PIE database contains 68 individuals with face images captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination and expression. As in [31, 52], we chose the five near frontal poses (C05, C07, C09, C27, C29) and used all the images under different illumination, lighting and expressions. Thus, there are about 170 samples per subject and there are a total number of 11,554 face images. Figure 1.7(a) shows 160 sample face images for a subject in this database.

Figure 1.8 shows the detailed face recognition results on the CMU PIE database for various values of L. The correct classification rates (CCRs) for each algorithm in comparison are plotted against the number of features used. To examine the discriminability of the most discriminative features extracted by each algorithm in detail, the horizontal axis (the number of features) is shown in log scale. The best results for each algorithm on the PIE database are reported in Table 1.2, where the best CCR for each L is highlighted with bold fonts.

From the detailed results, it can be seen that the first a few features extracted by the UMLDA algorithm consistently outperforms all the other algorithms, although the number of useful features extracted by UMLDA is limited compared to other MLDA variants [30]. In contrast, the heuristic TR1DA algorithm, built upon a greedy approximation approach, performs the worst in most cases, especially when the number of samples per subject is small (e.g., $L = 5, 10$). Similarly, the DATER algorithm outperforms the GTDA greatly on the PIE database. Thus, the MLDA variants based on scatter ratio have achieved much better results than the MLDA variants based on the scatter difference in this experiment, with the added benefit that there is no need to choose a tuning parameter ζ.

For the comparison between MLDA-TTP and MLDA-TVP, we focus on the scatter ratio-based variants: UMLDA and DATER. As mentioned above, the most discriminative features extracted by UMLDA seem to outperform the most discriminative features extracted by DATER. However, UMLDA has limited number of useful features in comparison. The results in Table 1.2 show that their performances are close on the PIE database.

Regarding the comparison between LDA and MLDA, we concentrate on the scatter-ratio-based methods: LDA and ULDA versus DATER and UMLDA. In this experiment, DATER and UMLDA outperform LDA and ULDA greatly, especially when L is small. When $L = 30$, i.e., the number of training samples for each subject is large, the performance gap is reduced. This comparison demonstrates that treating gray-level face images in their natural 2-D representation is advantageous against vectorized representation, especially when the number of training samples per subject is small.

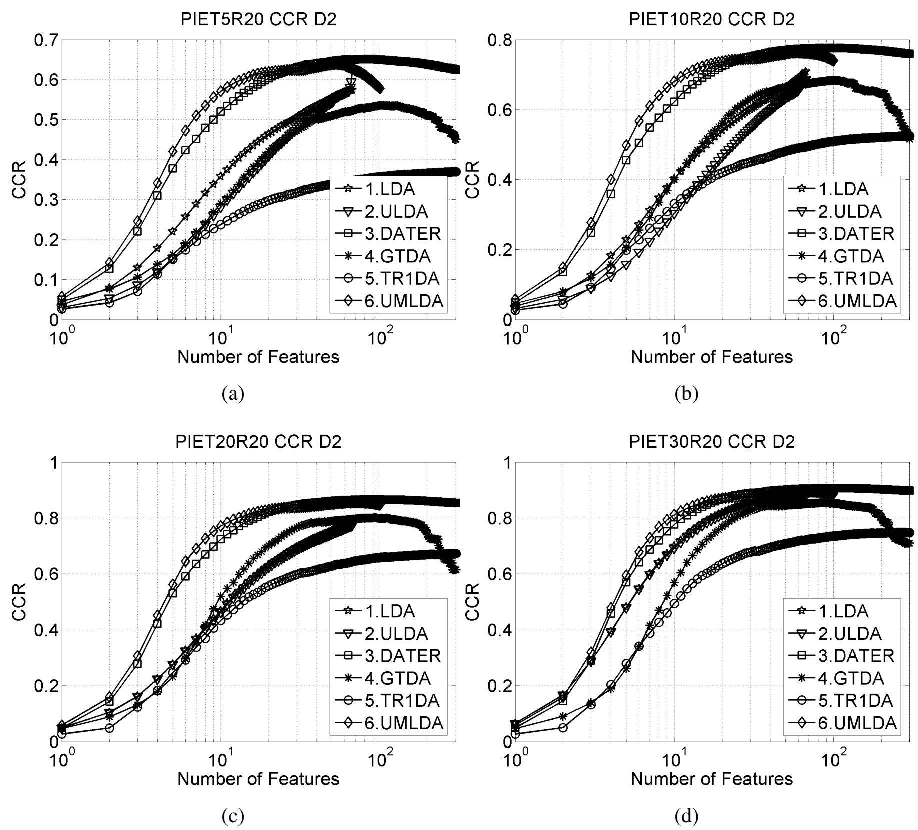

Fig. 1.8 Detailed face recognition results on the CMU PIE database with (a)$L = 5$, (b)$L = 10$, (c)$L = 20$, (d)$L = 30$.

1.4.3 Face recognition results on YaleB database

The Extended Yale face database B (YaleB) consists of 2,414 frontal face images of 38 individuals, which were captured under various laboratory-controlled lighting conditions. There are about 64 samples per subject and 60 sample face images for a subject are shown in Fig. 1.7(b).

Figure 1.9 shows the detailed face recognition results on the YaleB database, and the best results for each algorithm on the YaleB database are reported in Table 1.3, in a similar way as Section 1.4.2 for various values of L.

Table 1.2 Face recognition results on PIE database.

L	Fisherface(LDA)	ULDA	DATER	GTDA	TR1DA	UMLDA
5	0.574	0.626	**0.651**	0.537	0.370	0.639
10	0.708	0.684	**0.776**	0.684	0.525	0.763
20	0.785	0.774	**0.866**	0.801	0.672	0.856
30	0.891	0.880	**0.906**	0.856	0.749	0.894

From the detailed results in Fig. 1.9, it can be seen that the first a few features extracted by the UMLDA algorithm again consistently outperforms all the other algorithms. The heuristic TR1DA algorithm performs the worst for $L = 5, 10$. From Table 1.3, the DATER algorithm outperforms the GTDA slightly on the YaleB database but the performance gap is quite small. Overall, the MLDA variants based on scatter ratio again obtained better results than the MLDA variants based on scatter difference.

As in Section 1.4.2, we focus on the scatter-ratio-based variants, UMLDA and DATER, for the comparison between MLDA-TTP and MLDA-TVP. From Figure 1.9 and Table 1.3, UMLDA consistently outperforms DATER significantly on this database, especially for a smaller L, although the performance of UMLDA deteriorates when the number of features exceeds a certain number. Thus, on this database, the UMLDA, an MLDA-TVP approach extracting uncorrelated features, shows its advantage against the MLDA-TTP approach, where the features can be viewed to be extracted through interdependent EMPs.

Regarding the comparison between LDA and MLDA, there is an interesting observation from this experiment. For the Fisherface (LDA) approach, when L increases from 20 to 30, the recognition rate ironically decreases, as seen in Table 1.3. For the ULDA, when L increases from 10 to 20, the recognition rate surprisingly decreases too, as in Table 1.3. This is in contrary with our belief that more training samples should result in better recognition performance. On the other hand, all the four MLDA variants do not have this problem on this database, with recognition rate increasing as L increases, showing that MLDA approaches are more stable and consistent. Furthermore, the UMLDA algorithm outperforms the LDA and ULDA significantly, especially for a larger L, demonstrating again the benefits of extracting features directly from the natural 2-D representation of face images rather than from their vectorized representation.

1.4.4 Discussions

In summary, through the comparison in Figs 1.8 and 1.9, and Tables 1.2 and 1.3, it can be seen that by treating face images in their natural 2-D representation, the MLDA solution UMLDA achieves very good recognition results consistently on two very challenging face databases, for various number of training samples per subjects

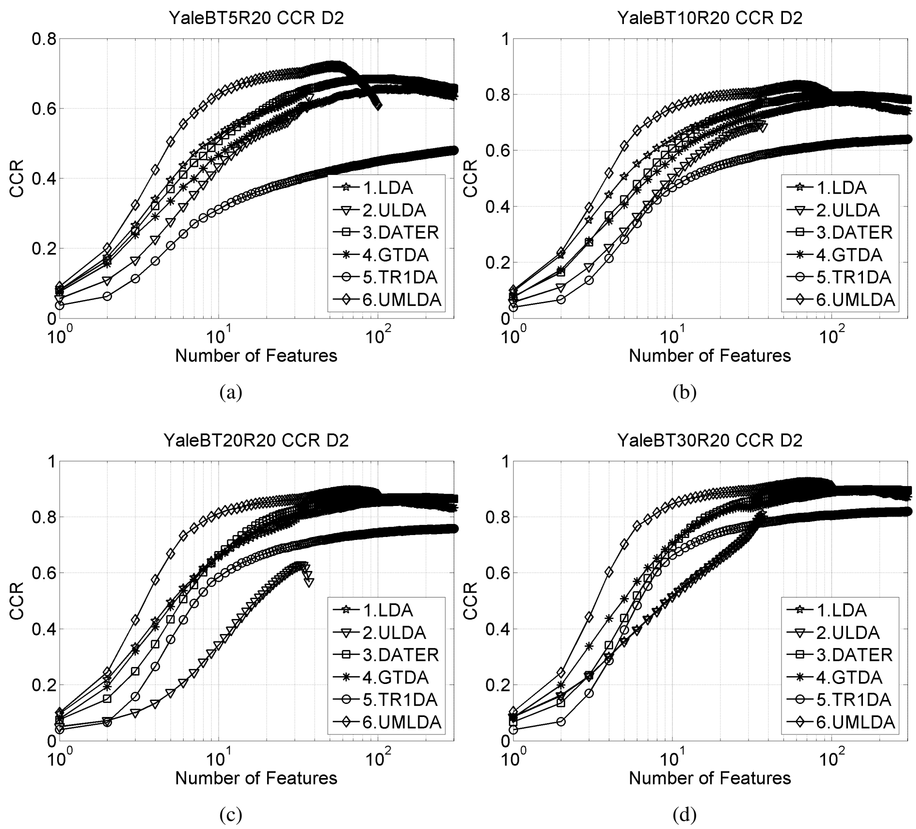

Fig. 1.9 Detailed face recognition results on the YaleB database with (a)$L = 5$, (b)$L = 10$, (c)$L = 20$, (d)$L = 30$.

Table 1.3 Face recognition results on YaleB database.

L	Fisherface(LDA)	ULDA	DATER	GTDA	TR1DA	UMLDA
5	0.653	0.632	0.685	0.657	0.480	**0.720**
10	0.783	0.695	0.797	0.777	0.640	**0.831**
20	0.858	0.628	0.870	0.856	0.758	**0.892**
30	0.812	0.792	0.900	0.894	0.819	**0.921**

($L = 5, 10, 20$ and 30). It is also observed that the MLDA variants based on scatter ratio generally outperform the MLDA variants based on scatter difference, and with scatter ratio as the separation criterion, the overall performance of MLDA-TVP is better than that of MLDA-TTP. In addition, MLDA variants are shown to be more stable and consistent than LDA approaches.

Considering the short period of research and development in multilinear learning solutions for biometric signal recognition, the empirical evaluation results presented here are very encouraging and we believe that there is still great potential in further development of multilinear learning algorithms that operate directly on natural tensorial representations. The materials provided in this chapter is a good starting point for newcomers to this field and the taxonomy of various multilinear projections and MLDA variants, together with discussions on their connections, is also beneficial for researchers already working in this field.

1.5 CONCLUSIONS

This chapter provides a comprehensive introduction to the area of multilinear learning algorithms, in particular the multilinear discriminant analysis (MLDA) algorithms, for the recognition of biometric signals, most of which are naturally tensor objects. Three typical projections are introduced first: the vector-to-vector projections (VVP), the tensor-to-tensor projections (TTP) and the tensor-to-vector projections (TVP), and two general MLDA solutions are formulated: the MLDA-TTP and the MLDA-TVP. The choices of the separation criteria and the initialization methods are then presented and the relationships between LDA, MLDA-TTP and MLDA-TVP are discussed. A taxonomy of MLDA variants is subsequently suggested and it not only helps us to understand the existing mutlilinear algorithms, but also benefits us in the development of new multilinear algorithms. Finally, the MLDA variants are experimentally evaluated on the CMU PIE database and the extended Yale database B to demonstrate their performance on the popular face recognition problem. The experimental results indicate that the MLDA solutions, and multilinear learning algorithms in general, are promising emerging areas for research and applications.

Acknowledgments

We would like to thank Cai Deng from the University of Illinois at Urbana-Champaign for making the standard-size face data available. We also would like to acknowledge all those who contributed to the research described in this chapter.

Appendix: Multilinear decompositions

There are two types of decompositions used most in multilinear applications: the canonical decomposition (CANDECOMP) [21, 22, 53], which is also known as the parallel factors (PARAFAC) decomposition [21, 22, 54], and the TUCKER decomposition [21, 22, 55].

With the CANDECOMP decomposition, a tensor $\mathcal{A}$ can be decomposed into a linear combination of P rank-1 tensors:

$$\mathcal{A} = \sum_{p=1}^{P} \lambda_p \mathbf{u}_p^{(1)} \circ \mathbf{u}_p^{(2)} \circ ... \circ \mathbf{u}_p^{(N)}, \tag{A.1}$$

where $P \leq \prod_{n=1}^{N} I_n$. With the TUCKER decomposition, a tensor $\mathcal{A}$ can be expressed as the product:

$$\begin{aligned} \mathcal{A} &= \sum_{p_1=1}^{P_1} \sum_{p_2=1}^{P_2} ... \sum_{p_N=1}^{P_N} \mathcal{S}(p_1, p_2, ..., p_N) \mathbf{u}_{p_1}^{(1)} \circ \mathbf{u}_{p_2}^{(2)} \circ ... \circ \mathbf{u}_{p_N}^{(N)} \\ &= \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times ... \times_N \mathbf{U}^{(N)}, \end{aligned} \tag{A.2}$$

where $P_n \leq I_n$ for $n = 1, ..., N$, $\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}$, and $\mathbf{U}^{(n)} = \left(\mathbf{u}_1^{(n)} \mathbf{u}_2^{(n)} ... \mathbf{u}_{P_n}^{(n)} \right)$ is an $I_n \times P_n$ matrix with orthonormal column vectors. The CONDECOMP decomposition is in fact a special case of the TUCKER decomposition.

References

1. L. D. Lathauwer, B. D. Moor, and J. Vandewalle, “On the best rank-1 and rank-$(R_1, R_2, ..., R_N)$ approximation of higher-order tensors,” *SIAM Journal of Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.

2. S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, “Multilinear discriminant analysis for face recognition,” *IEEE Trans. Image Processing*, vol. 16, no. 1, pp. 212–220, Jan. 2007.

3. H. Lu, J. Wang, and K. N. Plataniotis, “A review on face and gait recognition: System, data and algorithms,” in *Advanced Signal Processing Handbook*, 2nd ed., S. Stergiopoulos, Ed. Boca Raton, Florida: CRC Press, 2009, to appear.

4. Y.-D. Kim and S. Choi, “Color face tensor factorization and slicing for illumination-robust recognition,” in *Proc. Int. Conf. on Biometrics*, August 2007, pp. 19–28.

5. K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*. Springer Verlag, Berlin, 2000.

6. K. W. Bowyer, K. Chang, and P. Flynn, “A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition,” *Computer Vision and Image Understanding*, vol. 101, no. 1, pp. 1–15, Jan. 2006.

7. S. Z. Li, C. Zhao, X. Zhu, and Z. Lei, “3D+2D face recognition by fusion at both feature and decision levels,” in *Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*, Oct. 2005.

8. C. Liu and H. Wechsler, “Independent component analysis of gabor features for face recognition,” *IEEE Trans. Neural Networks*, vol. 14, no. 4, pp. 919–928, July 2003.

9. R. Chellappa, A. Roy-Chowdhury, and S. Zhou, *Recognition of Humans and Their Activities Using Video*. Morgan & Claypool Publishers, 2005.

10. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “A layered deformable model for gait analysis,” in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, Apr. 2006, pp. 249 – 254.

11. N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: a challenging signal processing technology for biometrics," *IEEE Signal Processing Mag.*, vol. 22, no. 6, Nov. 2005.

12. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A full-body layered deformable model for automatic model-based gait recognition," *EURASIP Journal on Advances in Signal Processing: Special Issue on Advanced Signal Processing and Pattern Recognition Methods for Biometrics*, vol. 2008, 2008, article ID 261317, 13 pages, doi:10.1155/2008/261317.

13. Z. Lei, R. Chu, R. He, S. Liao, and S. Z. Li, "Face recognition by discriminant analysis with gabor tensor representation," in *Proc. Int. Conf. on Biometrics*, August 2007, pp. 87–95.

14. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

15. G. Shakhnarovich and B. Moghaddam, "Face recognition in subspaces," in *Handbook of Face Recognition*, S. Z. Li and A. K. Jain, Eds. Springer-Verlag, 2004, pp. 141–168.

16. J. Zhang, S. Z. Li, and J. Wang, "Manifold learning and applications in recognition," in *Intelligent Multimedia Processing with Soft Computing*, Y. P. Tan, K. H. Yap, and L. Wang, Eds. Springer-Verlag, 2004, pp. 281–300.

17. M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 3, pp. 377–391, Mar. 2006.

18. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Gait recognition through MPCA plus LDA," in *Proc. Biometrics Symposium 2006*, September 2006, pp. 1–6, doi:10.1109/BCC.2006.4341613.

19. D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

20. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear principal component analysis through successive variance maximization," in *Proc. International Conference on Machine Learning*, July 2008, pp. 616–623.

21. L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singualr value decomposition," *SIAM Journal of Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

22. B. W. Bader and T. G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32, no. 4, pp. 635–653, Dec. 2006.

23. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Multilinear principal component analysis of tensor objects for recognition,” in *Proc. International Conference on Pattern Recognition*, vol. 2, August 2006, pp. 776 – 779.

24. D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, and H.-Y. Shum;, “Concurrent subspaces analysis,” in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. II, June 2005, pp. 203–208.

25. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Boosting LDA with regularization on MPCA features for gait recognition,” in *Proc. Biometrics Symposium 2007*, September 2007, doi:10.1109/BCC.2007.4430542.

26. J. Ye, R. Janardan, and Q. Li, “Two-dimensional linear discriminant analysis,” in *Advances in Neural Information Processing Systems (NIPS)*, 2004, pp. 1569–1576.

27. Y. Wang and S. Gong, “Tensor discriminant analysis for view-based object recognition,” in *Proc. Int. Conf. on Pattern Recognition*, vol. 3, August 2006, pp. 33 – 36.

28. D. Tao, X. Li, X. Wu, and S. J. Maybank, “Elapsed time in human gait recognition: A new approach,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 2, Apr. 2006, pp. 177 – 180.

29. T. G. Kolda, “Orthogonal tensor decompositions,” *SIAM Journal of Matrix Analysis and Applications*, vol. 23, no. 1, pp. 243–255, 2001.

30. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Uncorrelated multilinear discriminant analysis with regularization for gait recognition,” in *Proc. Biometrics Symposium 2007*, September 2007, doi:10.1109/BCC.2007.4430540.

31. X. He, D. Cai, and P. Niyogi, “Tensor subspace analysis,” in *Advances in Neural Information Processing Systems 18 (NIPS)*, 2005.

32. G. Dai and D. Y. Yeung, “Tensor embedding methods,” in *Proc. Twenty-First National Conference on Artificial Intelligence*, July 2006, pp. 330–335.

33. S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin, “Graph embedding and extensions: A general framework for dimensionality reduction,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

34. D. Xu, S. Lin, S. Yan, and X. Tang, “Rank-one projections with adaptive margins for face recognition,” *IEEE Trans. Syst., Man, Cybern. B*, vol. 37, no. 5, pp. 1226–1236, Oct. 2007.

35. G. Hua, P. A. Viola, and S. M. Drucker, “Face recognition using discriminatively trained orthogonal rank one tensor projections,” in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

36. L. D. Lathauwer and J. Vandewalle, “Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, ..., R_N)$ reduction in multilinear algebra,” *Linear Algebra and its Applications*, vol. 391, pp. 31–55, Nov. 2004.

37. H. Lu, A. C. Kot, and Y. Q. Shi, “Distance-reciprocal distortion measure for binary document images,” *IEEE Signal Processing Lett.*, vol. 11, no. 2, pp. 228–231, Feb. 2004.

38. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, Second Edition*. Wiley Interscience, 2001.

39. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 711–720, July 1997.

40. J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using lda based algorithms,” *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 195–200, Jan. 2003.

41. J. Han and B. Bhanu, “Individual recognition using gait energy image,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, no. 2, pp. 316–322, Feb. 2006.

42. J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Face recognition using kernel direct discriminant analysis algorithms,” *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 117–126, Jan. 2003.

43. T. K.Moon and W. C. Stirling, *Mathematical methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

44. Q. Liu, X. Tang, H. Lu, and S. Ma, “Face recognition using kernel scatter-difference-based discriminant analysis,” *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 1081–1085, July 2006.

45. Z. Liang, “High-dimensional discriminant analysis and its application to color face images,” in *Proc. Int. Conf. on Pattern Recognition*, vol. 2, August 2006, pp. 917 – 920.

46. D. Tao, X. Li, X. Wu, and S. J. Maybank, “Human carrying status in visual surveillance,” in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1670 – 1677.

47. H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, “Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition,” *IEEE Trans. Neural Networks*, 2009, to appear.

48. T. Sim, S. Baker, and M. Bsat, “The CMU pose, illumination, and expression database,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

49. A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 6, pp. 643–660, June 2001.

50. K. C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

51. J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *Journal of Machine Learning Research*, vol. 6, pp. 483–502, Apr. 2005.

52. D. Cai, X. He, J. Han, and H. J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Trans. Image Processing*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.

53. J. D. CARROLL and J. J. CHANG, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.

54. R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

55. L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.