Multilinear Subspace Learning for Face and Gait Recognition

by

Haiping Lu

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto

Your file Votre référence
ISBN: 978-0-494-58043-1
Our file Notre référence
ISBN: 978-0-494-58043-1

Multilinear Subspace Learning for Face and Gait Recognition

Haiping Lu

Doctor of Philosophy, 2008

Graduate Department of The Edward S. Rogers Sr. Department of Electrical and Computer Engineering

University of Toronto

Abstract

Face and gait recognition problems are challenging due to largely varying appearances, highly complex pattern distributions, and insufficient training samples. This dissertation focuses on multilinear subspace learning for face and gait recognition, where low-dimensional representations are learned directly from tensorial face or gait objects.

This research introduces a unifying multilinear subspace learning framework for systematic treatment of the multilinear subspace learning problem. Three multilinear projections are categorized according to the input-output space mapping as: vector-to-vector projection, tensor-to-tensor projection, and tensor-to-vector projection. Techniques for subspace learning from tensorial data are then proposed and analyzed. Multilinear principal component analysis (MPCA) seeks a tensor-to-tensor projection that maximizes the variation captured in the projected space, and it is further combined with linear discriminant analysis and boosting for better recognition performance. Uncorrelated MPCA (UMPCA) solves for a tensor-to-vector projection that maximizes the captured variation in the projected space while enforcing the zero-correlation constraint. Uncorrelated multilinear discriminant analysis (UMLDA) aims to produce uncorrelated features through

a tensor-to-vector projection that maximizes a ratio of the between-class scatter over the within-class scatter defined in the projected space. Regularization and aggregation are incorporated in the UMLDA solution for enhanced performance.

Experimental studies and comparative evaluations are presented and analyzed on the PIE and FERET face databases, and the USF gait database. The results indicate that the MPCA-based solution has achieved the best overall performance in various learning scenarios, the UMLDA-based solution has produced the most stable and competitive results with the same parameter setting, and the UMPCA algorithm is effective in unsupervised learning in low-dimensional subspace. Besides advancing the state-of-the-art of multilinear subspace learning for face and gait recognition, this dissertation also has potential impact in both the development of new multilinear subspace learning algorithms and other applications involving tensor objects.

Dedication

To my wife, Rong Xu.

Thanks for your love, support, understanding, and encouragement.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Konstantinos N. Plataniotis, for his guidance, nurturing, encouragement, and support in every stage of my graduate study. I have conceived numerous valuable ideas from frequent discussions with him. He has been a perfect role model for me to learn how to do research. Looking back at my graduate career, it has been a great fortune to have Prof. Plataniotis as my supervisor. His knowledge, kindness, patience, and vision have provided me with lifetime benefits. Also, I would like to thank my co-supervisor, Prof. Anastasios N. Venetsanopoulos, who has been always supportive in many aspects over all these years. From him, I have not only learned the knowledge of advanced image processing but also the leadership, which could be a huge fortune for my future career.

I am very grateful to Prof. Dimitrios Hatzinakos and Prof. Raymond H. Kwong for their insightful comments and suggestions on my thesis work. I would also like to express my sincere gratitude to Prof. Emil M. Petriu for his invaluable time to serve as an external examiner. It was a privilege for me to have each of them serve on my doctoral committee. In addition, I would like to thank all the members in the multimedia laboratory for their warm welcome and support. Special thanks go to my colleagues and friends, Dr. Juwei Lu, Dr. Jie Wang, Karl Martin, Azadeh Kushki, Yongjin Wang, Tejaswini Ganapathi, and Mohammad Shahin Mahanta. I have benefited a lot from their friendship and help. It has been really a great pleasure for me to study and work under the nice environment constructed and maintained by the efforts of all the members in the lab. I also thank all those anonymous reviewers of my papers for their constructive comments.

Special thanks to Prof. Yun-Qing Shi for guiding me into the research areas of image processing, Prof. Alex C. Kot for supervising me in my undergraduate and Master's studies and providing me with many valuable ideas and advices, Prof. Xudong Jiang for introducing me into the field of biometrics, and Dr. Nikolaos V. Boulgouris for sharing his experience in gait recognition with me.

Finally, I want to thank my parents, my two sisters, and my wife for their love and support, without which I could never accomplish all these achievements. Thanks to all the people who have ever helped and encouraged me.

Contents

List of Tables

List of Figures

List of Acronyms

Acronym	Description	Page #
2DLDA	Two-dimensional Linear Discriminant Analysis	55
2DPCA	Two-dimensional Principle Component Analysis	50
AdaBoost	Adaptive Boosting	80
ALS	Alternating Least Square	33
B-LDA-MPCA	Boosting LDA on MPCA features	80
CMC	Cumulative Match Characteristic	38
CRR	Correct Recognition Rate	39
CSA	Concurrent Subspaces Analysis	52
DATER	Discriminant Analysis with Tensor Representation	56
FERET	Facial Recognition Technology	41
GEI	Gait Energy Image	161
GLRAM	Generalized Low Rank Approximation of Matrices	52
GPCA	Generalized Principle Component Analysis	53
GTDA	General Tensor Discriminant Analysis	56
HMM	Hidden Markov Model	161
HOSVD	High Order Singular Value Decomposition	59
LDA	Linear Discriminant Analysis	8
LTN	Linear Time Normalization	161
MAD	Modified Angle Distance	39
ML1	Modified L_1 Distance	39
ML2	Modified L_2 Distance	39
MMD	Modified Mahalanobis Distance	39
MPCA	Multilinear Principal Component Analysis	11

Acronym	Description	Page #
PCA	Principle Component Analysis	8
PCDD	Pairwise Class Discriminant Distribution	84
PIE	Pose, Illumination, and Expression	41
R-UMLDA	Regularized UMLDA	117
R-UMLDA-A	Regularized UMLDA with Aggregation	127
SVD	Singular Value Decomposition	49
Std	Standard Deviations	142
TR1DA	Tensor Rank-One Discriminant Analysis	57
TROD	Tensor Rank-One Decomposition	54
ULDA	Uncorrelated Linear Discriminant Analysis	58
UMLDA	Uncorrelated Multilinear Discriminant Analysis	12
UMPCA	Uncorrelated Multilinear Principal Component Analysis	11

Important Notations

$a = 1, ..., A$	the index of the feature extractor in aggregation in R-UMLDA-A
A	the number of feature extractors to be aggregated in R-UMLDA-A
c	the class index
c_m	the class label for the mth training sample
C	the number of classes (subjects) in training
$H_{\mathbf{y}}$	the number of discriminative MPCA features for direct recognition or for input to LDA
I_n	the dimensionality of the n-mode
k	the iteration step index in multilinear solutions
K	the maximum number of iterations in multilinear solutions
L	the number of training samples for each class (subject)
M	the number of training samples in the gallery
M_c	the number of training samples in class c
n	the mode index of a tensor object
N	the order of a tensor object, the number of indices/modes
$p = 1, ..., P$	the index of the elementary multilinear projection in a tensor-to-vector projection
P	the number of elementary multilinear projections in a tensor-to-vector projection
P_n	the n-mode dimensionality in the projected space of a tensor-to-tensor projection
Q	the ratio of total scatter kept in each mode in MPCA
T	the number of boosting iterations

γ	the regularization parameter in R-UMLDA
κ	the regularization parameter in B-LDA-MPCA
ξ	the number of samples per class for LDA training in B-LDA-MPCA
ζ	the tuning parameter in scatter difference measures
$vec(\mathcal{A})$	the vectorized representation of the tensor $\mathcal{A}$
$\|\cdot\|_F$	the Frobenius norm
$\mathcal{X}_m$	the mth input tensor sample
$\tilde{\mathbf{U}}^{(n)}$	the nth projection matrix
$\mathcal{Y}_m$	the projection of $\mathcal{X}_m$ on $\{\tilde{\mathbf{U}}^{(n)}\}$
$\Psi_{\mathcal{Y}}$	the total tensor scatter of $\{\mathcal{Y}_m, m = 1, ..., M\}$
$\mathcal{Y}^*_{var}$	the total scatter tensor for full projection in MPCA
$\mathbf{u}^{(n)}$	the n-mode projection vector, $n = 1, ..., N$
$\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$	the pth elementary multilinear projection
$\mathbf{y}_m$	the vector rearranged from $\mathcal{Y}_m$ in MPCA the projection of $\mathcal{X}_m$ on $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ in UMPCA and UMLDA
$\mathbf{z}_m$	the projection of $\mathbf{y}_m$ in the LDA space in MPCA+LDA
$\mathbf{y}_m(p) = y_{m_p} = \mathbf{g}_p(m)$	the projection of $\mathcal{X}_m$ on $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$
$S^{\mathbf{y}}_{T_p}$	the total scatter of $\{y_{m_p}, m = 1, ..., M\}$
$S^{\mathbf{y}}_{B_p}$	the between-class scatter of $\{y_{m_p}, m = 1, ..., M\}$
$S^{\mathbf{y}}_{W_p}$	the within-class scatter of $\{y_{m_p}, m = 1, ..., M\}$
$\mathbf{g}_p$	the pth coordinate vector
$F^{\mathbf{y}}_p = \frac{S^{\mathbf{y}}_{B_p}}{S^{\mathbf{y}}_{W_p}}$	the Fisher's discrimination criterion for $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$
$\mathbf{D}_t(m, c)$	the mislabel distribution in AdaBoost.M2

Chapter 1

Introduction

This dissertation studies *multilinear subspace learning*, the problem of learning low-dimensional representations directly from tensorial data, for face and gait recognition [43, 87]. This chapter begins by introducing the field of biometrics, and in particular the problems of face and gait recognition. Next, the key challenges in solving these problems are outlined. It is then pointed out that for the popular subspace learning technique, traditional linear algorithms have their fundamental limitations and have become inadequate in handling the intrinsically tensorial facial and gait data, thus motivating the multilinear subspace learning approach. Finally, the contributions are listed and a road map to the rest of the dissertation is given.

1.1 Introduction to Biometrics

Biometrics refers to the automatic recognition of individuals based on their physiological and/or behavioral characteristics [43]. *Physiological* characteristics are related to the shape of the body, such as fingerprints, faces, hand geometry, and iris. *Behavioral* characteristics are related to the behavior of a person, such as signature, keystroke, voice, and gait. Although biometrics emerged from its extensive use in law enforcement to identify criminals, it is being increasingly used for human recognition in a large number of civilian

applications. Biometric system offers greater security and convenience than traditional methods of personal recognition, such as ID cards and passwords. It gives users greater convenience (e.g., no need to remember passwords) while maintaining sufficiently high accuracy and ensuring that the user is present at the point at time of recognition [42].

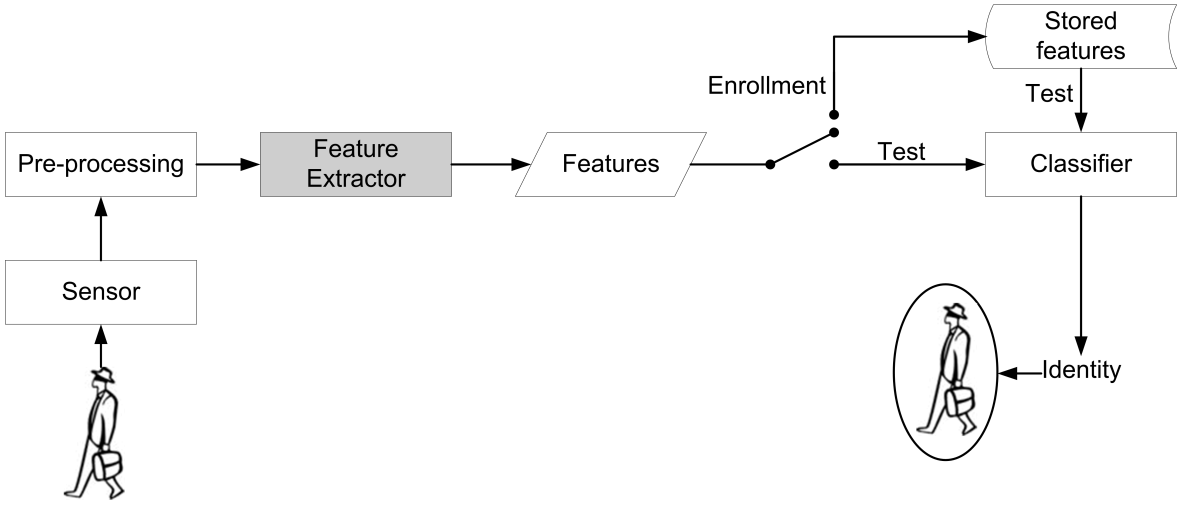


Figure 1.1: A general biometric system for human identification.

A biometric system is essentially a pattern recognition system that operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the stored feature sets in the database [43]. Figure 1.1 shows a simple block diagram of a typical biometric system. The main operations such a system can perform are enrollment and test. During the enrollment, biometric information from an individual is stored. During the test, biometric information is extracted and compared with the stored information. The sensor is the interface between the real world and the system, and it acquires all the necessary data. The second block performs all the necessary pre-processing, such as noise removal, enhancement, segmentation, and normalization. In the third block, features are extracted. This step is a critical step for successful recognition, and hence, it is the focus of this dissertation. In an enrollment, the features are simply stored. In a test, the features extracted from the input sample are passed to a classifier to classify against the stored features to determine

the identity.

1.2 Face and Gait Recognition

Face and gait are two typical physiological and behavioral biometrics, respectively. Compared with other biometric traits, face and gait have the unique property that they facilitate human recognition at a distance, which is extremely important in surveillance applications. Moreover, their unintrusive nature leads to high collectability and acceptability, making them very promising technologies for wide deployments. The *collectability* refers to the ease of acquisition for measurement. The *acceptability* indicates the extent to which people are willing to accept the use of a particular biometric identifier in their daily lives [43].

Face recognition has received significant attention during the past two decades, and it has a large number of commercial security and forensic applications, including video surveillance, access control, mugshot identification, entertainment industry, video communications, and medical diagnosis [43, 13].

Gait recognition is a relatively new area in biometrics. Gait, a person's walking style, is a complex spatio-temporal biometric [13, 98, 44]. It is considered to be the only true remote biometric [42]. The interest in gait recognition is strongly motivated by the need for an automated human identification system at a distance in visual surveillance and monitoring applications in security-sensitive environments, e.g., banks, parking lots, museums, malls, and transportation hubs such as airports and train stations [141]. Other biometrics such as fingerprint, face or iris information are usually not available at high enough resolution for recognition in these circumstances [13, 48]. Furthermore, night vision capability (an important component in surveillance) is usually not possible with other biometrics due to the limited biometric details in an IR image at large distance [13, 48]. Therefore, gait recognition, the identification of individuals through the way

they walk, has emerged as a promising solution with the advantages of unobtrusiveness, hard-to-hide, and recognition at a distance [49, 109, 4].

Face recognition algorithms are broadly categorized into appearance-based [128, 3, 114, 89, 151, 91] and feature-based [108, 143, 17] algorithms and they take digitally captured facial image, mostly in gray-level, as input. Gait recognition algorithms are either appearance-based [71,135,68,141,126,25,8,31] or model-based [140,149,18,136], and their inputs are usually binary gait silhouette sequences since color or texture is not reliable for recognition. In both face and gait recognition, the appearance-based algorithms are arguably the most successful ones [10,160,67,119,99,94] and they have support from studies in visual neuroscience [119]. Therefore, this dissertation focuses on *appearance-based learning*[1], where the input images or image sequences are treated as holistic patterns.

1.3 Challenges in Appearance-Based Face and Gait Recognition

Although extensive studies have been performed and encouraging progresses have been made in appearance-based face and gait recognition, these two problems remain largely unsolved. The study and development of a comprehensive learning framework for face and gait signals has both theoretical and practical implications of broad significance. The key technical challenges pertinent to this research are summarized in the following:

1. **Large variability of the appearance:** Face or gait patterns of the same person generally exhibit significant variations in appearance [88]. The intra-subject variations for face patterns include pose (imaging angle), illumination, facial expression, occlusion, makeup, glasses, facial hair, time (aging), and imaging parameters such as aperture and exposure time. The intra-subject variations for gait patterns in-

[1]This research has also investigated a model-based approach for gait recognition, with a layered deformable model proposed [78, 82].

clude pose (viewing angle), shoes, walking surface, carrying condition, clothing, time, and also imaging device. Such intra-subject variations can be larger than the variations due to the change of subject identities. This makes the extraction of discriminative information from a face or gait object a demanding task.

2. **High complexity of pattern distribution:** It is commonly believed that face or gait patterns of the same subject under the large number of variations lie in a nonlinear manifold. The class conditional distribution of the face or gait patterns is generally believed to be multi-modal and non-convex [88]. This severely challenges the existing pattern recognition methodologies that often assume much simpler distribution, and makes the face or gait recognition task extremely difficult.

3. **Insufficiency of training samples:** Face or gait patterns for recognition purposes usually have high dimensionality. For example, the size of a facial image in typical recognition applications ranges from 32×32 [37] to 150×130 pixels [93], which correspond to dimensionality of 1,024 to 19,500 pixels. The size of a typical gait sequence for recognition ranges from $32 \times 22 \times 10$ to $128 \times 88 \times 20$ pixels, which correspond to dimensionality of 7,040 to 225,280 pixels. However, in practical face and gait recognition applications, the number of samples (per subject) available for training is often much smaller than the number of parameters to be estimated, causing the so-called *small sample size* problem. The lack of adequate training samples significantly degrades the performance of the feature extractors and the classifiers, especially in supervised learning [88].

From the above discussions, the problems of face and gait recognition present researchers with both challenges and opportunities. Hence, they are examined in this dissertation with emphasis on the feature extraction module of Fig. 1.1.

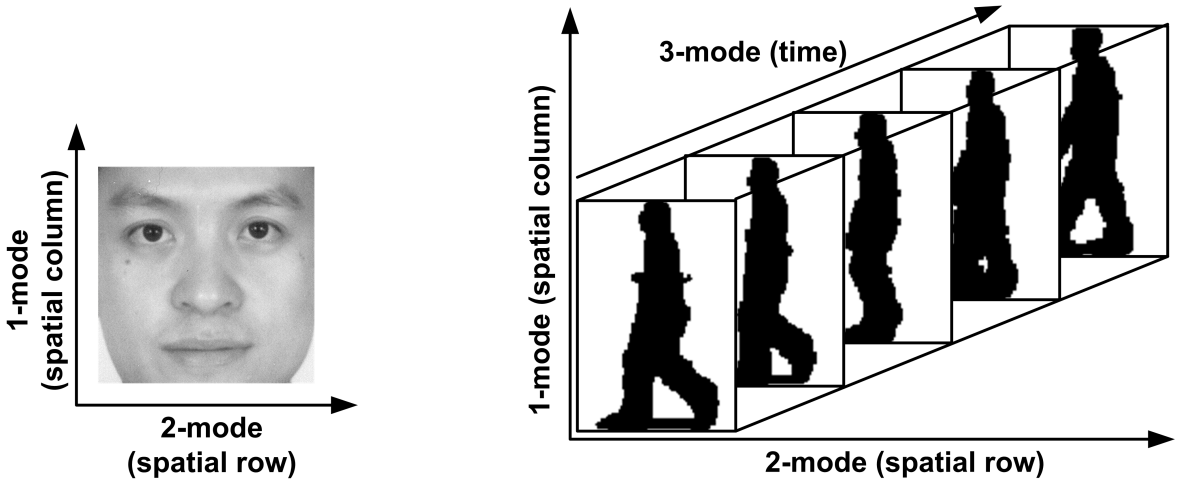
1.4 Motivation: the Approach of Multilinear Subspace Learning

This dissertation takes the approach of multilinear subspace learning in solving the face and gait recognition problems. Before describing this approach, the concept of tensor object is introduced and the natural representations of face and gait objects are studied first.

1.4.1 The natural representations of face and gait objects

In this dissertation, tensorial data, or multi-dimensional objects/arrays, are formally referred to as *tensor objects*. The elements of a *tensor* are to be addressed by N indices, where N (the number of indices used in the description) defines the *order* of the tensor object and each index defines one *mode* [57,60]. By this definition, vectors are first-order tensors (with $N = 1$) and matrices are second-order tensors (with $N = 2$). Tensors with $N > 2$ can be viewed as a generalization of vectors and matrices to higher order. A tensor object is an element in a *tensor space*, the *Kronecker product* (also known as *direct product* or *tensor product*) of N vector spaces [33, 57, 96]. In addition, it should be noted that the term tensor has different meanings in mathematics and physics. The usage in this dissertation refers to its meaning in mathematics, in particular multilinear algebra [59, 60, 33, 57]. In physics, the same term usually means the so-called *tensor field* [63], a generalization of the vector field. It is an association of a different tensor with each point of a geometric space and it varies continuously with position.

By the definitions above, gray-level face images are naturally second-order tensors with the column, and row modes [150], and binary gait silhouette sequences are third-order tensors with the column, row, and time modes. For illustration, Fig. 1.2 shows two examples in their natural representations with their modes labeled: a gray-level face image in Fig. 1.2(a) and a binary gait silhouette sequence in Fig. 1.2(b).



(a) (b)

Figure 1.2: The natural representations of two biometric objects: (a) a second-order face tensor, and (b) a third-order gait (silhouette) tensor.

Besides gray-level images and binary/gray-level video sequences, there are other real-world tensor objects as well. Color images are third-order tensors with the column, row, and color modes [51]. Three-dimensional gray-level objects [19], such as 3-D gray-level faces [7, 66], are naturally third-order tensors with the column, row, and depth modes. Color video sequences are fourth-order tensors with the column, row, time, and color modes.

1.4.2 Multilinear subspace learning

As mentioned in Sec. 1.3, a face or gait object is commonly specified in a high-dimensional space. Recognition methods operating directly on this space suffer from the so-called curse of dimensionality [115]: handling high-dimensional samples is computationally expensive and many classifiers perform poorly in high-dimensional spaces given a small number of training samples. However, face or gait objects do not lie randomly in the high-dimensional space, rather, they are highly constrained and confined to a subspace, a manifold of intrinsically low dimension [115, 159]. *Subspace learning* (also known as *dimensionality reduction*) is thus an attempt to transform a high-dimensional data set

into a low-dimensional equivalent representation while retaining most of the information regarding the underlying structure or the actual physical phenomenon [62]. It is the arguably most successful approach in appearance-based learning.

Traditional subspace learning algorithms, such as the classical principal component analysis (PCA) [47] and linear discriminant analysis (LDA) [21], are linear algorithms that operate on one-dimensional objects, i.e., first-order tensors (vectors). PCA reduces the dimensionality of the data by retaining most of the variation in the input data through a linear projection that produces uncorrelated features [47]. LDA maximizes a ratio of the between-class scatter to the within-class scatter in order to best separate classes. To apply these linear algorithms to higher-order (greater than one) tensor objects, such as images and videos, these tensor objects have to be reshaped (vectorized) into vectors first. However, such reshaping, i.e., vectorization, usually results in very high dimensional vectors that lead to high or even impractical computational and memory demands, in particular for massive data sets such as video sequences. It also requires the estimation of a large number of parameters, which is often a problem, especially in the small sample size scenario, where the number of training samples available is limited. Furthermore, the vectorization breaks the natural structure and correlation in the original data, reduces redundancies and/or higher order dependencies present in the original data set, and loses potentially more compact or useful representations that can be obtained in the original tensorial forms. Thus, multilinear subspace learning, subspace learning algorithms operating directly on the tensor objects rather than their vectorized versions, are desirable and they offer great potential in processing tensor objects.

In the past a few years, a number of multilinear subspace learning algorithms have been proposed [147,156,116,153,37,155,150,124,142,123,37,20,151,144,41]. Despite the encouraging progress made, multilinear subspace learning is still a field in its infancy. It remains to be a very difficult problem to successfully extend the rich ideas developed in the linear subspace learning to multilinear subspace learning.

Firstly, subspace learning relies on projection from one (high-dimensional) space to another (low-dimensional) space. Compared to the well-understood linear projection, multilinear projection is a relatively new concept in subspace learning. Although multilinear projection has been used in existing multilinear subspace learning algorithms, there is no work giving a formal, systematic treatment on this topic, which hinders the development of multilinear subspace learning.

Secondly, linear subspace learning algorithms such as PCA and LDA are not iterative. However, in multilinear subspace learning, there are usually N sets of parameters to be estimated, with one set in each mode. The estimation of parameters in one mode usually depends on the parameters in all the other modes. Hence, an iterative procedure has to be adopted to solve for all parameters. In turn, the issues of initialization, projection order, termination, and convergence, which do not exist in linear subspace learning, have to be addressed in multilinear subspace learning.

Thirdly, the classical PCA and LDA algorithms both derive uncorrelated features, i.e., features with zero correlation. Uncorrelated features contain minimum redundancy and ensure linear independence of features. They can greatly simplify the subsequent classification task and they are highly desirable in many applications [158]. However, there is no existing multilinear subspace learning algorithm producing uncorrelated features due to the difficulty in constraint enforcement in a multilinear setting.

Lastly, although the small sample size problem is reduced in multilinear subspace learning as the number of parameters to be estimated is much smaller in multilinear subspace learning than in linear subspace learning, this number in supervised multilinear subspace learning still far exceeds the number of samples available for their accurate estimation in most practical situations. Thus, the small sample size problem needs to be tackled as well in supervised multilinear subspace learning.

1.5 Contributions

In light of the proceeding discussion, this dissertation aims to advance the state-of-the-art in the area of multilinear subspace learning for face and gait recognition. The central thesis of this dissertation is that, face and gait recognition algorithms, both supervised and unsupervised in nature, can be developed within a unifying multilinear subspace learning framework through the appropriate incorporation of iterative solutions and acceptable learning constraint.

Figure 1.3 depicts in the form of a "tree" the major contributions made in this research. These novel contributions, shown as "shaded boxes" in Figure 1.3, are explained below in a top-to-down and left-to-right order:

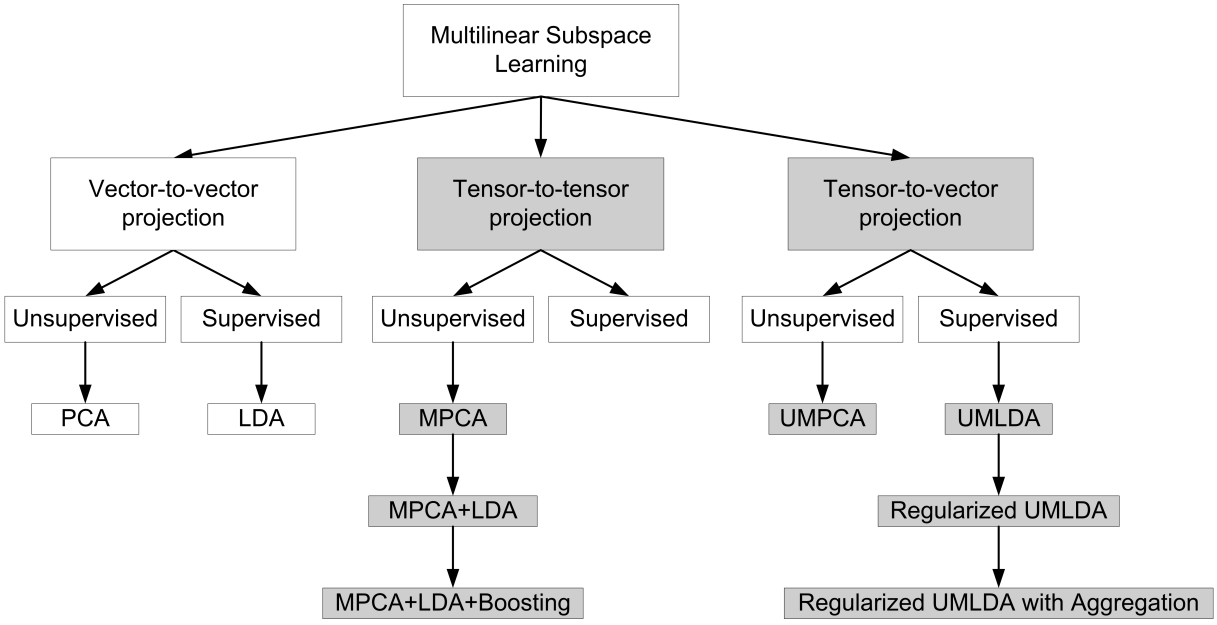


Figure 1.3: The contributions (shaded boxes) of this dissertation in multilinear subspace learning for face and gait recognition.

1. **A systematic treatment on multilinear projection:** The dissertation systematically treats the notion of multilinear projection [86], upon which all the developed multilinear subspace learning algorithms are based. The three different multilinear projections discussed in this work are categorized according to the input-output

space mapping. Namely, the vector-to-vector projection, the tensor-to-tensor projection, and the tensor-to-vector projection are introduced, analyzed, and commented upon. This unified representation allows for the systematic treatment and comparative evaluation of the various existing multilinear subspace learning algorithms. Moreover, it facilitates the development of novel multilinear subspace learning algorithms.

2. **Multilinear principal component analysis (MPCA):** The dissertation introduces the MPCA algorithm [83], an unsupervised multilinear subspace learning algorithm developed using the tensor-to-tensor projection. The MPCA algorithm extends the idea of variance maximization in PCA to tensorial input data. MPCA seeks a tensor-to-tensor projection that maximizes the total tensor scatter in the projected space. This is the first unsupervised multilinear subspace learning algorithm with the objective of variance maximization that can be applied to general tensorial data. This research addresses issues related to the algorithm initialization phase, projection order determination, termination criterion, and convergence properties. It also proposes solutions for the problem of subspace dimensionality determination, which is important in algorithms based on the tensor-to-tensor projection. Furthermore, enhanced variants of MPCA are developed by discriminative feature selection and by its combination with LDA and boosting [83, 80, 77]. To the best of the author's knowledge, the proposed MPCA+LDA+boosting approach is the first known work that combines ensemble-based learning with a multilinear subspace solution.

3. **Uncorrelated multilinear principal component analysis (UMPCA):** The dissertation introduces, researches, and evaluates the so-called UMPCA algorithm [85], an unsupervised multilinear subspace learning algorithm using the tensor-to-vector projection. In UMPCA, a zero-correlation constraint on the produced

features is enforced in addition to seeking a tensor-to-vector projection that maximizes the total scalar-based scatter in the projected space. UMPCA is motivated by the fact that in addition to variance maximization, PCA derives uncorrelated features. Thus, UMPCA extends PCA through the tensor-to-vector projection and it is the first known unsupervised multilinear subspace learning algorithm that produces uncorrelated features.

4. **Uncorrelated multilinear discriminant analysis (UMLDA):** The dissertation introduces the so-called UMLDA algorithm [84], a supervised multilinear subspace learning algorithm using the tensor-to-vector projection. As in PCA, the classical LDA algorithm also derives uncorrelated features. UMLDA maximizes the class separability in the projected space, as measured by the traditional scatter ratio, while enforcing the zero-correlation constraint among extracted features. Therefore, UMLDA extends the classical LDA through the tensor-to-vector projection and it is the first known supervised multilinear subspace learning algorithm that produces uncorrelated features. Moreover, as mentioned in Sec. 1.3, supervised learning algorithms are more susceptible to the small sample size problem in practice. Hence, the recognition performance of UMLDA is further enhanced through regularization and aggregation.

1.6 Organization

The rest of the dissertation is organized as follows.

Chapter 2 introduces the fundamentals of multilinear subspace learning. Basic multilinear algebra is reviewed first. Three basic types of multilinear projections are then discussed: the vector-to-vector projection (linear projection), the tensor-to-tensor projection, and the tensor-to-vector projection. Furthermore, their connections and differences are analyzed. Next, the problem of multilinear subspace learning is defined along with the

tensor-based and scalar-based measures for the total scatter, the between-class scatter, and the within-class scatter. In brief, this chapter offers a systematic treatment of the fundamental concepts needed in the development of new multilinear subspace learning algorithms in this dissertation.

Chapter 3 provides an overview of the performance evaluation mechanism and the data sets used in experimental evaluation. It also offers a comprehensive review of the existing state-of-the-art multilinear subspace learning algorithms. The commonly used terminology is introduced and the performance evaluation schemes, which will be used throughout this dissertation, are presented in detail. This chapter describes the face and gait databases used in the experiments and outlines the respective preprocessing steps. Next, existing multilinear subspace learning algorithms are reviewed in detail, highlighting their limitations and pitfalls.

Chapter 4 introduces the MPCA algorithm and its extensions. The problem to be solved is defined and an iterative solution is derived to maximize the captured variance through the tensor-to-tensor projection. Connections with the existing solutions are discussed. Issues pertinent to the initialization, projection order, termination, and convergence of the algorithm are examined in detail. Methods for subspace dimensionality determination are proposed. To improve recognition performance, a discriminative MPCA feature selection procedure is suggested, and the MPCA+LDA algorithm is also proposed. The chapter further introduces the combination of MPCA and the LDA-style booster in [93] for better generalization performance. Three synthetic data sets with different eigenvalue distributions are generated, on which the MPCA properties regarding the initialization, projection order, convergence, and subspace dimensionality determination are studied in detail.

Chapter 5 presents the UMPCA algorithm. UMPCA aims to extract uncorrelated features through the tensor-to-vector projection while maximizing the variance in the projected space. The solution follows the successive variance maximization approach

in a classical derivation of PCA. The tensor-to-vector projection is viewed as a number of elementary projections and these elementary projections are solved one by one, sequentially, with each step being iterative as in MPCA. The limitation of UMPCA in the number of features that can be extracted is theoretically analyzed. The connections of UMPCA with other algorithms are pointed out and design issues similar to those in MPCA are discussed. Since UMPCA is also unsupervised, its properties regarding initialization, projection order, and convergence are studied on the three synthetic data sets constructed in Chapter 4.

Chapter 6 proposes the UMLDA algorithm, which is supervised. UMLDA aims to extract uncorrelated features through the tensor-to-vector projection while maximizing the scatter ratio in the projected space. Furthermore, because the small sample size problem is more severe for supervised learning algorithms, regularization are adopted to enhance its performance in practical situations. The solution for UMLDA is derived in a similar fashion as that for UMPCA. UMLDA is also limited in the number of extracted features and an aggregation scheme is proposed to overcome this limitation. Since UMLDA is a supervised algorithm, its properties regarding initialization, regularization, projection order, convergence, and aggregation are studied on a face database so that labeled data is available for supervised learning and the performance under different sample sizes can be examined.

Chapter 7 evaluates the face and gait recognition performance of the proposed algorithms by comparing them against existing state-of-the-art subspace learning algorithms. Six sets of experiments are introduced. The first and second sets of face recognition experiments evaluate the algorithms under varying number of training samples per class and varying number of classes, respectively. The third set of face recognition experiments conducts specific studies of the unsupervised learning algorithms and in particular examines the advantage of UMPCA in low-dimensional subspace. The first set of gait recognition experiments evaluates the proposed and existing subspace learning algorithms

under various capturing conditions. The second set of gait recognition experiments compares the MPCA-based algorithms against the state-of-the-art gait recognition algorithms with more sophisticated preprocessing and matching algorithms. The third set of gait recognition experiments carries out the first known boosting studies on gait recognition. The experimental results are discussed and important observations are made.

Chapter 8 concludes this dissertation by summarizing the key contributions and suggesting directions for future research.

The technical contents of Chapters 2, 4, 6, and 7 have partly appeared in the following IEEE copyrighted materials, with the permission to reprint granted by IEEE:

1. Haiping Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects", *IEEE Transactions on Neural Networks*, Vol. 19, No. 1, Page: 18-39, January 2008.

2. Haiping Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition", revision submitted to *IEEE Transactions on Neural Networks*, 2008 (accepted pending minor revision).

3. Haiping Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A taxonomy of emerging multilinear discriminant analysis solutions for biometric signal recognition", to appear in *Biometrics: Theory, Methods, and Applications*, N. Boulgouris, K.N. Plataniotis, and E. Micheli-Tzanakou, Eds., IEEE/Wiley Press (submitted in 2007, currently under review).

4. Haiping Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Boosting LDA with regularization on MPCA features for gait recognition", in *Proceedings of the Biometrics Symposium 2007* (BSYM 2007), Baltimore, US, September 2007.

Chapter 2

Fundamentals of Multilinear Subspace Learning

This chapter formulates the general problem of multilinear subspace learning. It begins by introducing the notations and reviewing basic multilinear algebra. A commonly used tensor distance measure is then shown to be equivalent to the Euclidean distance for corresponding vectors. Next, three basic types of multilinear projections, including linear projection, are formulated and their underlying connections with each other are analyzed. After a brief review of two classical linear subspace learning methods, the problem of multilinear subspace learning is defined, together with several tensor-based and scalar-based scatter measures. Finally, the typical approach in solving multilinear problems is discussed. With a systematic approach, this chapter serves as the foundations for the rest of this dissertation, and it helps the readers to understand multilinear concepts with ease and clarity for usage and even further development of multilinear subspace learning algorithms.

2.1 Multilinear Basics

This section reviews the notations and some basic multilinear operations [60, 59, 2] that are necessary in defining the multilinear subspace learning problem. To pursue further in multilinear algebra, [33, 57, 59, 60, 61, 53, 2] are excellent references. In addition, the equivalent vector interpretation of a commonly used tensor distance measure is derived in this section.

2.1.1 Notations

The notations in this dissertation follow the conventions in the multilinear algebra, pattern recognition, and adaptive learning literature [60, 59, 2]. 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 parentheses. Indices are denoted by lowercase letters and span the range from 1 to the uppercase letter of the index, e.g., $n = 1, 2, ..., N$. To address part of a vector/matrix/tensor, ":" denotes the full range of the corresponding index and $n_1 : n_2$ denotes indices ranging from n_1 to n_2. Throughout this dissertation, the discussion is restricted to real-valued vectors, matrices, and tensors since the targeted applications involve real-valued data only, such as gray-level face images and binary gait silhouette sequences.

2.1.2 Basic multilinear algebra

As in [60, 59, 2], 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:

$$(\mathcal{A} \times_n \mathbf{U})(i_1, ..., i_{n-1}, j_n, i_{n+1}, ..., i_N) = \sum_{i_n} \mathcal{A}(i_1, ..., i_N) \cdot \mathbf{U}(j_n, i_n). \tag{2.1}$$

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:

$$< \mathcal{A}, \mathcal{B} >= \sum_{i_1} \sum_{i_2} ... \sum_{i_N} \mathcal{A}(i_1, i_2, ..., i_N) \cdot \mathcal{B}(i_1, i_2, ..., i_N) \tag{2.2}$$

and the Frobenius norm of $\mathcal{A}$ is defined as

$$\| \mathcal{A} \|_F = \sqrt{< \mathcal{A}, \mathcal{A} >}. \tag{2.3}$$

The i_nth "n-mode slice" of $\mathcal{A}$ is an $(N-1)$th-order tensor obtained by fixing the n-mode index of $\mathcal{A}$ to be i_n: $\mathcal{A}(:, ..., :, i_n, :, ..., :)$. 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:

$$\mathcal{A} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}, \tag{2.4}$$

which means that

$$\mathcal{A}(i_1, i_2, ..., i_N) = \mathbf{u}^{(1)}(i_1) \cdot \mathbf{u}^{(2)}(i_2) \cdot ... \cdot \mathbf{u}^{(N)}(i_N) \tag{2.5}$$

for all values of indices. Unfolding $\mathcal{A}$ along the n-mode is denoted as

$$\mathbf{A}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times ... \times I_{n-1} \times I_{n+1} \times ... \times I_N)}, \tag{2.6}$$

and the column vectors of $\mathbf{A}_{(n)}$ are the n-mode vectors of $\mathcal{A}$.

Following standard multilinear algebra, any tensor $\mathcal{A}$ can be expressed as the product:

$$\mathcal{A} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times ... \times_N \mathbf{U}^{(N)}, \tag{2.7}$$

where

$$\mathcal{S} = \mathcal{A} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}, \tag{2.8}$$

and $\mathbf{U}^{(n)} = \left(\mathbf{u}_1^{(n)} \mathbf{u}_2^{(n)} ... \mathbf{u}_{I_n}^{(n)}\right)$ is an orthogonal $I_n \times I_n$ matrix. Since $\mathbf{U}^{(n)}$ has orthonormal columns, $\| \mathcal{A} \|_F^2 = \| \mathcal{S} \|_F^2$ [60]. A matrix representation of this decomposition can be obtained by unfolding $\mathcal{A}$ and $\mathcal{S}$ as

$$\mathbf{A}_{(n)} = \mathbf{U}^{(n)} \cdot \mathbf{S}_{(n)} \cdot \left(\mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n+2)} \otimes ... \otimes \mathbf{U}^{(N)} \otimes \mathbf{U}^{(1)} \otimes \mathbf{U}^{(2)} \otimes ... \otimes \mathbf{U}^{(n-1)}\right)^T, \tag{2.9}$$

where $\bigotimes$ denotes the Kronecker product. The decomposition can also be written as:

$$\mathcal{A} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} ... \sum_{i_N=1}^{I_N} \mathcal{S}(i_1, i_2, ..., i_N) \mathbf{u}_{i_1}^{(1)} \circ \mathbf{u}_{i_2}^{(2)} \circ ... \circ \mathbf{u}_{i_N}^{(N)}, \tag{2.10}$$

i.e., any tensor $\mathcal{A}$ can be written as a linear combination of $I_1 \times I_2 \times ... \times I_N$ rank-1 tensors.

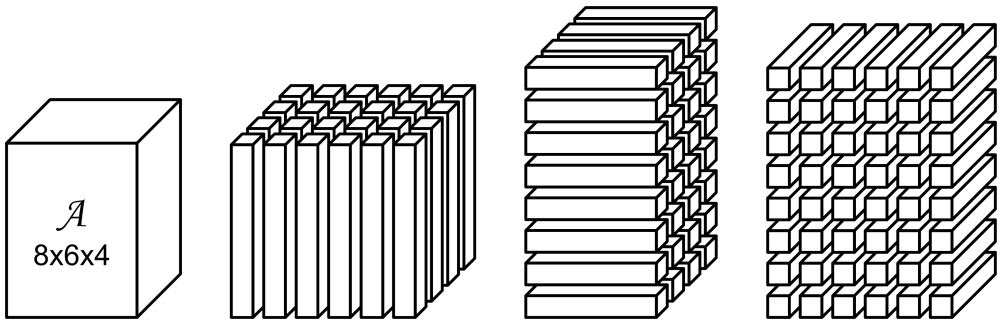
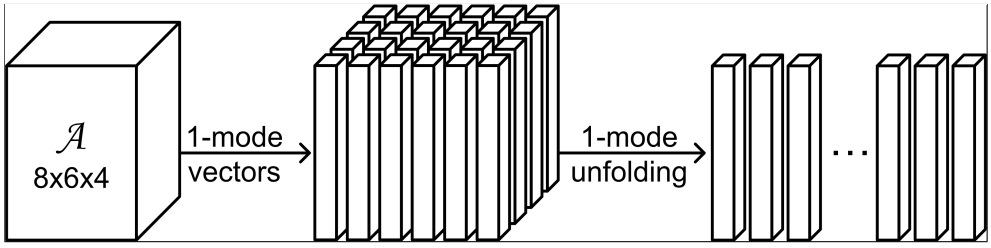
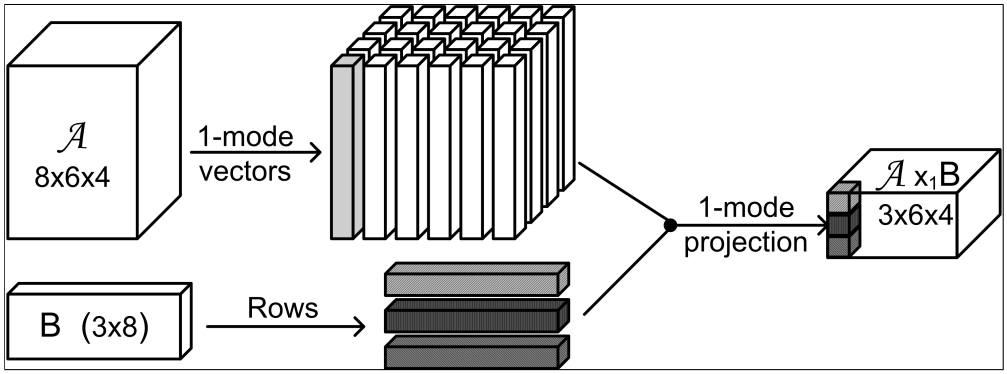


Figure 2.1: Illustration of the n-mode vectors: (a) a tensor $\mathcal{A} \in \mathbb{R}^{8 \times 6 \times 4}$, (b) the 1-mode vectors, (c) the 2-mode vectors, and (d) the 3-mode vectors.

Figures 2.1(b), 2.1(c), and 2.1(d) give visual illustrations of the 1-mode, 2-mode and 3-mode vectors of the third-order tensor $\mathcal{A}$ in Fig. 2.1(a), respectively. Figure 2.2(a) shows the 1-mode unfolding of the tensor $\mathcal{A}$ in Fig. 2.1(a). Fig. 2.2(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



(a)



(b)

Figure 2.2: Visual illustration of (a) the n-mode (1-mode) unfolding and (b) the n-mode (1-mode) multiplication.

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. 2.2(b).

2.1.3 Tensor distance measure

To measure the distance between tensors $\mathcal{A}$ and $\mathcal{B}$, the Frobenius norm is used in [150]:

$$dist(\mathcal{A}, \mathcal{B}) = \| \mathcal{A} - \mathcal{B} \|_F \,. \tag{2.11}$$

Although this is a tensor-based measure, it can be proven to be equivalent to a distance measure of corresponding vector representations. Let $vec(\mathcal{A})$ be the vector representation (vectorization) of $\mathcal{A}$, a property regarding the inner product between two tensors is

derived as follows:

Proposition 2.1. $< \mathcal{A}, \mathcal{B} >=< vec(\mathcal{X}), vec(\mathcal{U}) >= [vec(\mathcal{U})]^T vec(\mathcal{X})$.

Proof. From (2.2),

$$\begin{aligned} < \mathcal{A}, \mathcal{B} > &= \sum_{i_1} \sum_{i_2} ... \sum_{i_N} \mathcal{A}(i_1, i_2, ..., i_N) \cdot \mathcal{B}(i_1, i_2, ..., i_N) \\ &= \sum_{i=1}^{\prod_{n=1}^{N} I_n} vec(\mathcal{A})(i) \cdot vec(\mathcal{B})(i) \\ &= < vec(\mathcal{A}), vec(\mathcal{B}) > \\ &= [vec(\mathcal{B})]^T vec(\mathcal{A}). \end{aligned}$$

□

Then, it is straightforward to show that

Proposition 2.2. $dist(\mathcal{A}, \mathcal{B}) =\| vec(\mathcal{A}) - vec(\mathcal{B}) \|_2$.

Proof. From Proposition 2.1,

$$\begin{aligned} dist(\mathcal{A}, \mathcal{B}) &= \| \mathcal{A} - \mathcal{B} \|_F \\ &= \sqrt{< (\mathcal{A} - \mathcal{B}), (\mathcal{A} - \mathcal{B}) >} \\ &= \sqrt{< vec(\mathcal{A}) - vec(\mathcal{B}), vec(\mathcal{A}) - vec(\mathcal{B}) >} \\ &= \| vec(\mathcal{A}) - vec(\mathcal{B}) \|_2 . \end{aligned}$$

□

Proposition 2.2 indicates that the Frobenius norm of the difference between two tensors equals to the Euclidean distance between their vectorized representations. Another explanation is that the Frobenius norm is a point-based measurement as well [75] and it does not take the structure of a tensor into account.

2.2 Multilinear Projections

A multilinear subspace is defined through a multilinear projection that maps the input data from one space to another (lower-dimensional) space [37]. Therefore, what is a multilinear projection needs to be understood before proceeding to the multilinear subspace learning solutions.

This section first proposes a categorization of the three basic multilinear projections in terms of the input and output of the projection: the traditional vector-to-vector projection, the tensor-to-tensor projection, and the tensor-to-vector projection. Furthermore, the relationships between these projections are investigated.

2.2.1 Vector-to-vector projection

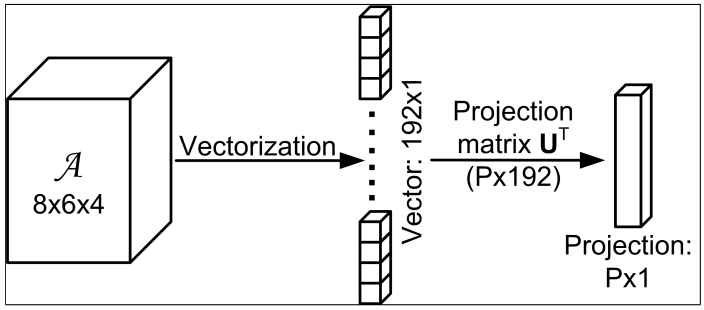
Linear projection is a standard transform used widely in various applications [21, 96]. A linear projection takes a vector $\mathbf{x} \in \mathbb{R}^I$ and projects it to $\mathbf{y} \in \mathbb{R}^P$ using a projection matrix $\mathbf{U} \in \mathbb{R}^{I \times P}$:

$$\mathbf{y} = \mathbf{U}^T\mathbf{x} = \mathbf{x} \times_1 \mathbf{U}^T. \tag{2.12}$$

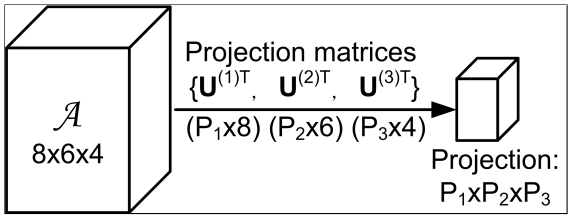
In typical pattern recognition applications, $P << I$. Therefore, linear projection is a vector-to-vector projection and it requires the vectorization of an input before projection. Figure 2.3(a) illustrates the vector-to-vector projection of a tensor object $\mathcal{A}$.

2.2.2 Tensor-to-tensor projection

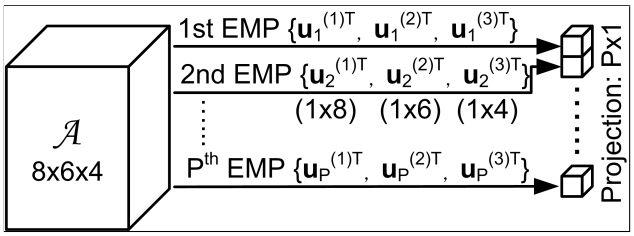
Besides the traditional vector-to-vector projection, a tensor can also be projected to another tensor (of the same order), named as the tensor-to-tensor projection in this chapter. An Nth-order tensor $\mathcal{X}$ resides in the tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$ [59, 37]. Thus, the tensor space 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 $P_n \leq I_n$



(a)



(b)



(c)

Figure 2.3: Illustration of (a) vector-to-vector projection, (b) tensor-to-tensor projection, (c) tensor-to-vector projection, where EMP stands for elementary multilinear projection.

for all n, N projection matrices $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, ..., N\}$ are used so that [60]

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} ... \times_N \mathbf{U}^{(N)^T}. \tag{2.13}$$

Figure 2.3(b) demonstrates the tensor-to-tensor projection 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. 2.2(b).

2.2.3 Tensor-to-vector projection

Next, a third multilinear projection is introduced, which is from a tensor space to a vector space, and it is called the tensor-to-vector projection[1]. The tensor-to-vector projection projects a tensor to a vector and it can be viewed as multiple projections from a tensor to a scalar, as illustrated in Fig. 2.3(c), where the tensor-to-vector projection of a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$ to a $P \times 1$ vector consists of P projections from $\mathcal{A}$ to a scalar. Thus, the projection from a tensor to a scalar is considered first.

A tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$ can be projected to a point y through N unit projection vectors $\{\mathbf{u}^{(1)^T}, \mathbf{u}^{(2)^T}, ..., \mathbf{u}^{(N)^T}\}$ as:

$$y = \mathcal{X} \times_1 \mathbf{u}^{(1)^T} \times_2 \mathbf{u}^{(2)^T} ... \times_N \mathbf{u}^{(N)^T}, \ \| \mathbf{u}^{(n)} \| = 1 \text{ for } n = 1, ..., N, \tag{2.14}$$

where $\| \cdot \|$ is the Euclidean norm for vectors. It can be written in the scalar product (2.2) as:

$$y = < \mathcal{X}, \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)} > . \tag{2.15}$$

Denote $\mathcal{U} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, then $y = < \mathcal{X}, \mathcal{U} >$. This multilinear projection $\{\mathbf{u}^{(1)^T}, \mathbf{u}^{(2)^T}, ..., \mathbf{u}^{(N)^T}\}$ is named as an elementary multilinear projection, the projection of a tensor on a single line (resulting a scalar), and it consists of one projection vector in each mode. Figure 2.4 illustrates an elementary multilinear projection of a tensor $\mathcal{A} \in \mathbb{R}^{8\times6\times4}$.

Thus, the tensor-to-vector projection of a tensor object $\mathcal{X}$ to a vector $\mathbf{y} \in \mathbb{R}^P$ in a P-dimensional vector space consists of P elementary multilinear projections

$$\{\mathbf{u}_p^{(1)^T}, \mathbf{u}_p^{(2)^T}, ..., \mathbf{u}_p^{(N)^T}\}, p = 1, ..., P, \tag{2.16}$$

which can be written concisely as $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^P$. The tensor-to-vector projec-

[1]The tensor-to-vector projection is referred to as the rank-one projections in some works [142,123,41].

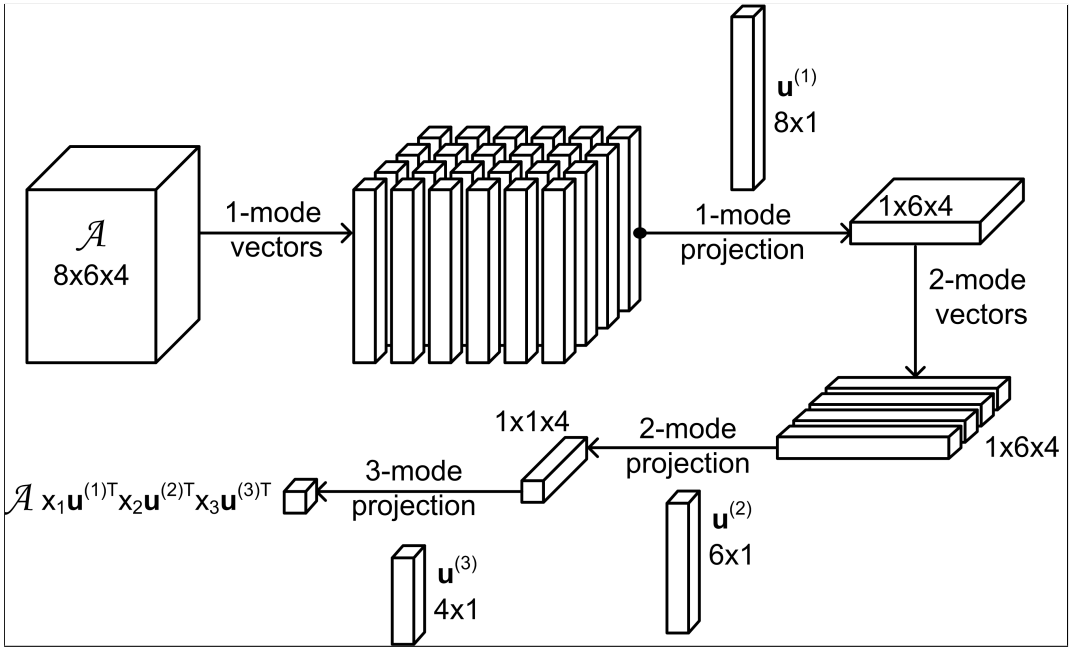


Figure 2.4: Illustration of an elementary multilinear projection.

tion from $\mathcal{X}$ to $\mathbf{y}$ is then written as

$$\mathbf{y} = \mathcal{X} \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}, \tag{2.17}$$

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

$$\mathbf{y}(p) = \mathcal{X} \times_1 \mathbf{u}_p^{(1)^T} \times_2 \mathbf{u}_p^{(2)^T} ... \times_N \mathbf{u}_p^{(N)^T}. \tag{2.18}$$

Figure 2.3(c) shows the tensor-to-vector projection of a tensor object $\mathcal{A}$ to a vector of size $P \times 1$.

2.2.4 Relationships between the three types of multilinear projections

With the introduction of the three basic multilinear projections, it is worthwhile to investigate their relationships. It is easy to verify that the vector-to-vector projection is

the special case of the tensor-to-tensor projection and the tensor-to-vector projection with $N = 1$. The elementary multilinear projection is the degenerated version of the tensor-to-tensor projection with $P_n = 1$ for all n. On the other hand, each projected element in the tensor-to-tensor projection can be viewed as the projection of an elementary multilinear projection formed by taking one column from each of the projection matrices. Thus, the projected tensor in the tensor-to-tensor projection is obtained through $\prod_{n=1}^{N} P_n$ interdependent elementary multilinear projections in effect, while in the tensor-to-vector projection, the P elementary multilinear projections obtained sequentially are not interdependent generally.

Furthermore, recall that the projection using an elementary multilinear projection $\{\mathbf{u}^{(1)^T}, \mathbf{u}^{(2)^T}, ..., \mathbf{u}^{(N)^T}\}$ can be written as

$$y =< \mathcal{X}, \mathcal{U} >=< vec(\mathcal{X}), vec(\mathcal{U}) >= [vec(\mathcal{U})]^T vec(\mathcal{X}), \tag{2.19}$$

by Proposition 2.1. Thus, an elementary multilinear projection is equivalent to a linear projection of $vec(\mathcal{X})$, the vectorized representation of $\mathcal{X}$, on a vector $vec(\mathcal{U})$. Since $\mathcal{U} = \mathbf{u}^{(1)} \circ \mathbf{u}^{(2)} \circ ... \circ \mathbf{u}^{(N)}$, (2.19) indicates that the elementary multilinear projection 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 the vector-to-vector projection specified by I parameters ($I = \prod_{n=1}^{N} I_n$ for an Nth-order tensor), an elementary multilinear projection in the tensor-to-vector projection can be specified by $\sum_{n=1}^{N} I_n$ parameters. Hence, to project a tensor of size $\prod_{n=1}^{N} I_n$ to a vector of size $P \times 1$, the tensor-to-vector projection needs to estimate only $P \cdot \sum_{n=1}^{N} I_n$ parameters, while the vector-to-vector projection needs to estimate $P \cdot \prod_{n=1}^{N} I_n$ parameters. The implication in pattern recognition problem is that the tensor-to-vector projection has fewer parameters to estimate while being more constrained on the solutions, and the vector-to-vector projection has less

constraint on the solutions sought while having more parameters to estimate.

2.3 Linear Subspace Learning

Linear subspace learning algorithms [21, 115] solve for a linear projection with some optimality criteria, given a set of training samples. The problem can be formulated mathematically as follows.

Linear Subspace Learning: A set of M vectorial samples $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M\}$ is available for training, where each sample $\mathbf{x}_m$ is an $I \times 1$ vector in a vector space $\mathbb{R}^I$. The linear subspace learning objective is to find a linear transformation (projection) $\mathbf{U} \in \mathbb{R}^{I \times P}$ such that the projected samples (the extracted features) $\{\mathbf{y}_m = \mathbf{U}^T \mathbf{x}_m\}$ satisfy an optimality criterion, where $\mathbf{y}_m \in \mathbb{R}^{P \times 1}$ and $P < I$. In classification, these features are fed into a classifier, e.g., the nearest neighbor classifier, and the similarity is usually calculated based on some distance measure.

Among various linear subspace learning algorithms, PCA [47] and LDA [21] are the two most important and widely used algorithms in a broad range of applications [128, 3]. PCA is an unsupervised method that does not require the training samples to be labeled, while LDA is a supervised method that makes use of class specific information. The algorithms developed in this dissertation (and many other subspace learning algorithms) are based on these two highly influential techniques, so they are reviewed below.

2.3.1 Principal component analysis

PCA is one of the most influential linear subspace learning methods. The well-known eigenface method [128] for face recognition, built on PCA, started the era of the appearance-based approach to face recognition, and more generally to visual object recognition. The central idea behind PCA is to reduce the dimensionality of a data set consisting of a larger number of interrelated variables, while retaining as much as possible the variation

present in the original data set [47]. This is achieved by transforming to a new set of variables, the so-called principal components, which are uncorrelated, and ordered so that the first few retain most of the original data variation. Thus, PCA aims to derive the most descriptive features.

In practice, the variation to be maximized is measured by the total scatter through the total scatter matrix $\mathbf{S}_T$ defined as follows,

$$\mathbf{S}_T = \sum_{m=1}^{M} (\mathbf{x}_m - \bar{\mathbf{x}})(\mathbf{x}_m - \bar{\mathbf{x}})^T, \tag{2.20}$$

where $\bar{\mathbf{x}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}_m$ is the mean of all the training samples. The PCA projection matrix $\mathbf{U}_{PCA}$ is then composed of the eigenvectors corresponding to the largest P ($P < I$) eigenvalues of $\mathbf{S}_T$. The projection of a test sample $\mathbf{x}$ in the PCA space is obtained as:

$$\mathbf{y} = \mathbf{U}_{PCA}^T(\mathbf{x} - \bar{\mathbf{x}}). \tag{2.21}$$

2.3.2 Linear discriminant analysis

LDA is a classical supervised linear subspace learning method that has been very successful and applied widely in various applications [21]. It aims to derive the most discriminative features and produces a class-specific feature space based on the maximization of the so-called Fisher's discriminant criterion [21, 3], defined as the ratio of between-class scatter to within-class scatter:

$$\mathbf{U}_{LDA} = \arg\max_{\mathbf{U}} \frac{|\mathbf{U}^T\mathbf{S}_B\mathbf{U}|}{|\mathbf{U}^T\mathbf{S}_W\mathbf{U}|}, \tag{2.22}$$

where $\mathbf{S}_B$ and $\mathbf{S}_W$ are the between-class and within-class scatter matrices, respectively, and they are defined as

$$\mathbf{S}_B = \sum_{c=1}^{C} M_c(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})(\bar{\mathbf{x}}_c - \bar{\mathbf{x}})^T, \tag{2.23}$$

and

$$\mathbf{S}_W = \sum_{m=1}^{M} (\mathbf{x}_m - \bar{\mathbf{x}}_{c_m})(\mathbf{x}_m - \bar{\mathbf{x}}_{c_m})^T. \tag{2.24}$$

In the definitions above, C is the number of classes, c is the class index, and c_m is the class label for the mth training sample. M_c is the number of training samples in class c, and the mean for class c is

$$\bar{\mathbf{x}}_c = \frac{1}{M_c} \sum_{m, c_m = c} \mathbf{x}_m. \tag{2.25}$$

The maximization of (2.22) leads to the following generalized eigenvalue problem:

$$\mathbf{S}_B \mathbf{u}_p = \lambda_p \mathbf{S}_W \mathbf{u}_p. \tag{2.26}$$

Thus, $\mathbf{U}_{LDA}$ consists of the generalized eigenvectors corresponding to the largest P generalized eigenvalues of (2.26). When $\mathbf{S}_W$ is not singular, $\mathbf{U}_{LDA}$ can be obtained as the eigenvectors corresponding to the largest P eigenvalues of $\mathbf{S}_W^{-1}\mathbf{S}_B$. The projection of a test sample $\mathbf{x}$ in the LDA space is then obtained as:

$$\mathbf{y} = \mathbf{U}_{LDA}^T \mathbf{x}. \tag{2.27}$$

In practice, LDA often has problem due to insufficient number of training samples, which results in singular $\mathbf{S}_W$. PCA is routinely employed before LDA to reduce the dimensionality before LDA to avoid this difficulty, leading to the PCA+LDA approach originally proposed in [3] for face recognition (the Fisherface method).

2.4 Multilinear Subspace Learning

This section defines the problem of multilinear subspace learning, as well as the scatter measures for tensors and scalars. In addition, the typical approach to solving such problems, together with related issues, is outlined.

2.4.1 Problem definition

Multilinear subspace learning is the multilinear extension of linear subspace learning. It solves for a multilinear projection with some optimality criteria, given a set of training samples. This problem can be formulated mathematically as follows, similar to the formulation in Sec. 2.3 for the linear subspace learning.

Multilinear Subspace Learning: A set of M Nth-order tensorial samples $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_M\}$ is available for training, where each sample $\mathcal{X}_m$ is an $I_1 \times I_2 \times ... \times I_N$ tensor in a tensor space $\mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$. The multilinear subspace learning objective is to find a multilinear transformation (projection) such that the projected samples (the extracted features) satisfy an optimality criterion, where the dimensionality of the projected space is much lower than the original tensor space. In classification, these features are fed into a classifier, e.g., the nearest neighbor classifier, and the similarity is calculated according to some distance measure.

At this point, the illustration in Fig. 1.3 (page 10) can be better appreciated. The projection to be solved can be any of the three types of basic multilinear projections discussed in Sec. 2.2. Thus, the well-studied linear subspace learning can be viewed as a special (degenerated) case of multilinear subspace learning where the projection to be solved is the vector-to-vector projection. The problem of multilinear subspace learning based on the tensor-to-tensor and tensor-to-vector projections is the focus of this dissertation. The formulation here is important for the purposes of evaluating, comparing, and further developing multilinear subspace learning solutions.

2.4.2 Tensor scatter

In analogy to the definition of scatters (2.20), (2.23), and (2.24) for vectorial features in linear subspace learning, tensor-based scatters in multilinear subspace learning are defined here.

Definition 2.1. *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 total scatter of these tensors is defined as:*

$$\Psi_{\mathcal{A}} = \sum_{m=1}^{M} \parallel \mathcal{A}_m - \bar{\mathcal{A}} \parallel_F^2, \tag{2.28}$$

where $\bar{\mathcal{A}}$ *is the mean tensor calculated as*

$$\bar{\mathcal{A}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{A}_m. \tag{2.29}$$

The n*-mode total scatter matrix of these samples is then defined as:*

$$\mathbf{S}_{T_{\mathcal{A}}}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)}\right) \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{(n)}\right)^T, \tag{2.30}$$

where $\mathbf{A}_{m(n)}$ *is the* n*-mode unfolded matrix of* $\mathcal{A}_m$.

Definition 2.2. *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:*

$$\Psi_{B_{\mathcal{A}}} = \sum_{c=1}^{C} M_c \parallel \bar{\mathcal{A}}_c - \bar{\mathcal{A}} \parallel_F^2, \tag{2.31}$$

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

$$\Psi_{W_{\mathcal{A}}} = \sum_{m=1}^{M} \parallel \mathcal{A}_m - \bar{\mathcal{A}}_{c_m} \parallel_F^2, \tag{2.32}$$

where C *is the number of classes,* M_c *is the number of samples for class* c, c_m *is the class label for the* m*th sample* $\mathcal{A}_m$, $\bar{\mathcal{A}}$ *is the mean tensor, and the class mean tensor is*

$$\bar{\mathcal{A}}_c = \frac{1}{M_c} \sum_{m, c_m = c} \mathcal{A}_m. \tag{2.33}$$

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

Definition 2.3. *The n-mode between-class scatter matrix of these samples is defined as:*

$$\mathbf{S}_{B_{\mathcal{A}}}^{(n)} = \sum_{c=1}^{C} M_c \cdot \left(\bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)}\right)\left(\bar{\mathbf{A}}_{c(n)} - \bar{\mathbf{A}}_{(n)}\right)^T, \tag{2.34}$$

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

$$\mathbf{S}_{W_{\mathcal{A}}}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)}\right)\left(\mathbf{A}_{m(n)} - \bar{\mathbf{A}}_{c_m(n)}\right)^T, \tag{2.35}$$

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

2.4.3 Scalar scatter

While the tensor scatters defined in the previous section are useful for developing multilinear subspace learning algorithms based on the tensor-to-tensor projections, they are not applicable for those based on the tensor-to-vector projections. Therefore, scalar-based scatters in multilinear subspace learning are defined, which can be viewed as the degenerated versions of the vector-based or tensor-based scatters.

Definition 2.4. *Let $\{a_m, m = 1, ..., M\}$ be a set of M scalar samples. The total scatter of these scalars is defined as:*

$$S_T^{\mathbf{a}} = \sum_{m=1}^{M} (a_m - \bar{a})^2, \tag{2.36}$$

where $\bar{a}$ is the mean scalar calculated as $\bar{a} = \frac{1}{M}\sum_{m=1}^{M} a_m$.

Definition 2.5. *Let $\{a_m, m = 1, ..., M\}$ be a set of M scalar samples. The between-class*

scatter of these scalars is defined as:

$$S_B^{\mathbf{a}} = \sum_{c=1}^{C} M_c(\bar{a}_c - \bar{a})^2, \tag{2.37}$$

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

$$S_W^{\mathbf{a}} = \sum_{m=1}^{M} (a_m - \bar{a}_{c_m})^2, \tag{2.38}$$

where $\bar{a}_c = \frac{1}{M_c} \sum_{m,c_m=c} a_m$.

2.4.4 Typical approach

While a linear (vector-to-vector) projection in linear subspace learning often has closed-form solutions, this is not true for the tensor-to-tensor and tensor-to-vector projections in multilinear subspace learning. Instead, these two tensor-based projections have N sets of parameters to be solved, one in each mode, and the solution to one set often depends on the other sets (except when $N = 1$, the linear case), making their simultaneous estimation extremely difficult, if not impossible. Therefore, a suboptimal, iterative procedure originated from the alternating least square (ALS) algorithm [12,35,56] is usually employed to solve the tensor-based projections by alternating between solving one set of parameters (in one mode) at a time. Consequently, the issues due to the iterative nature of the solution, such as the initialization, the order of solving the projections, the termination, and convergence, need to be addressed. In addition, for multilinear subspace learning through the tensor-to-tensor projection, a mechanism is often needed to determine the desired subspace dimensionality $\{P_1, P_2, ..., P_N\}$. This is because it is usually costly to exhaustively test the large number of possible combinations of the N values, $P_1, P_2, ..., P_N$, for a specific amount of dimensionality reduction. In contrast, only one value P needs to be tested for multilinear subspace learning through the tensor-to-vector

projection.

A brief review of the ALS algorithm is given here. The ALS algorithm was first developed in 1970 [35, 12] to solve a similar problem in the three-way factor analysis [23] where parameters in three modes need to be estimated. The principle behind the ALS is to reduce the (least square) optimization problem into smaller conditional subproblems that can be solved through simple established methods employed in the linear case. Thus, the parameters for each mode are estimated in turn separately and are conditional on the parameter values for the other modes. At each step, by fixing the parameters in all the modes but one mode, a new objective function depending only on the mode left free to vary is optimized and this conditional subproblem is linear and much simpler. The parameter estimations for each mode are obtained in this way sequentially and iteratively until convergence. A typical procedure for multilinear subspace learning is shown in Fig. 2.5.

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$, and the desired tensor subspace dimensionality.

Output: The multilinear projection that maximizes an optimality criterion in the projected space.

Algorithm:

Step 1 (Initialization): Initialize the multilinear projection.

Step 2 (Local optimization):

- For $k = 1 : K$
  - For $n = 1 : N$
    - Solve for the n-mode multilinear projection
  - If $k = K$ or the algorithm converges, break and output the current multilinear projection.

Figure 2.5: The pseudo-code of a typical multilinear subspace learning algorithm.

Table 2.1: Linear versus multilinear subspace learning.

Linear subspace learning	Multilinear subspace learning
Reshape into vectors	Work directly on natural tensorial representation
Break natural structure	Preserve natural structure
Estimate a large number of parameters	Estimate fewer parameters
More severe small sample size problem	Less small sample size problem
Hardly applicable to massive data	Able to handle massive data
Closed-form solution	Suboptimal, iterative solution

Finally, Table 2.1 summarizes the key differences between multilinear subspace learning and linear subspace learning. In the table, massive data refers to the data with its dimensionality beyond the processing power of common computational hardwares when linear subspace learning algorithms are used, such as face images with very high resolution or standard gait silhouette sequences.

2.5 Summary

This chapter has introduced a general formulation of the multilinear subspace learning problem. The fundamentals are covered, including basic multilinear operations and a tensor-based distance measure. Three basic types of multilinear projections are then introduced: the vector-to-vector projection, the tensor-to-tensor projection, and the tensor-to-vector projection. Moreover, the connections between these three projections are revealed. After a brief review of the classical linear subspace learning algorithms, the definition of the multilinear subspace learning problem is presented. This definition provides a framework that not only helps to explain many key aspects of multilinear subspace learning, but also facilitates the developments of new multilinear subspace learning algorithms. Lastly, several scatter measures are defined for tensors and scalars, and the typical approach to solve multilinear subspace learning problem is described.

The next chapter will review the background materials including recognition per-

formance evaluation, face and gait databases, and prior work on multilinear subspace learning. In particular, the multilinear subspace learning literature will be viewed in the framework developed in this chapter. In subsequent chapters, solutions are proposed in this research to show how different criteria and constraints can be incorporated to build effective algorithms for tensorial face and gait recognition.

Chapter 3

Review on Prior Work, Performance Evaluation & Data

In the advancement of face and gait recognition technologies, the availability of evaluation methodology, and large, representative, and public databases plays an important role besides the development of recognition algorithms. This chapter starts by discussing issues related to recognition performance evaluation. Next, this chapter reviews the databases to be used in this research to evaluate the developed learning algorithms, including two widely used face databases and one popular gait database. Finally, prior work on multilinear subspace learning are studied.

3.1 Recognition Performance Evaluation

In typical pattern recognition problems of face and gait recognition, there are usually two types of data sets: the *gallery* and the *probe* [102, 109]. The **gallery** set contains the set of data samples with known identities and it is used for training. The **probe** set is the testing set where data samples of unknown identity are to be identified and classified via matching with corresponding entries in the gallery set.

There are three main recognition tasks in face and gait recognition applications: *veri-*

fication, *identification*, and *watch list* [44]. **Verification** involves a one-to-one match that compares a query sample against the sample(s) of the claimed identity in the database. The claim is either accepted or rejected. The verification performance is usually measured by the receiver operating characteristic (ROC), which plots the false accept rates (FAR) versus the false rejection rates (FRR). **Identification** involves one-to-many matches that compare a query sample of an unknown person against the samples of all the persons in the database to output the identity or the possible identity list of the input query sample. In this scenario, it is often assumed that the unknown (query) person belongs to the persons who are in the database. The identification performance is usually measured by the cumulative match characteristic (CMC) [102, 109], which plots the identification rate R_ρ against the rank ρ. The **watch list** scenario involves one-to-few matches that compare a query sample against a list of suspects. In this task, the size of database is usually very small compared to the possible queries, and the identity of the probe may not be in the database. Therefore, the recognition system should first detect whether the query is on the list or not and if yes, correctly identify it. Correspondingly, the performance of watch list tasks is usually measured by the detection rate, the identification rate, and the false alarm rate.

This research focuses on the identification task and throughout this dissertation, the term recognition refers to the application scenario of identification. Based on the definitions above, the general face or gait recognition problem is stated as follows:

The face or gait recognition problem: Given a gallery database, consisting of face or gait samples from a set of known subjects, the objective of the face or gait recognition system is to determine the identity of the probe samples or sample sequences (with unknown identities). In short, the task of face or gait recognition is to determine the gallery subject to which a probe sample or sequence corresponds. The performance of both face recognition and gait recognition will be measured by the identification rate.

In the following, the computation of the identification rate is described in detail first,

adapted from [102]. Then, the calculation of similarity scores for both individual samples and sequences, where each sequence consists of several samples, is presented.

Computation of identification rate: Let $\mathbb{P} = \{\mathcal{P}_1, ..., \mathcal{P}_J\}$ be a probe set and J be the number of samples or sequences in $\mathbb{P}$. The gallery set $\mathbb{G} = \{\mathcal{G}_1, ..., \mathcal{G}_M\}$ has M samples or sequences. The probe set $\mathbb{P}$ is scored against the gallery set $\mathbb{G}$ by computing the similarity scores $S(\mathcal{P}_j, \mathcal{G}_m)$ for $\mathcal{P}_j \in \mathbb{P}$ and $\mathcal{G}_m \in \mathbb{G}$. For each probe sample or sequence $\mathcal{P}_j \in \mathbb{P}$, $S(\mathcal{P}_j, \mathcal{G}_m)$ is sorted for all gallery samples or sequences $\mathcal{G}_m \in \mathbb{G}$, where a higher similarity score implies a closer match. The function $id(j)$ gives the index of the gallery sample or sequence of the person in the probe sample or sequence $\mathcal{P}_j$, i.e., $\mathcal{P}_j$ is a sample or sequence of the person in $\mathcal{G}_{id(j)}$. A probe $\mathcal{P}_j$ is correctly identified if $S(\mathcal{P}_j, \mathcal{G}_{id(j)})$ is the highest scores for $\mathcal{G}_m \in \mathbb{G}$. A probe $\mathcal{P}_j$ is in the top ρ if $S(\mathcal{P}_j, \mathcal{G}_{id(j)})$ is one of the ρ highest score $S(\mathcal{P}_j, \mathcal{G}_m)$ for the gallery $\mathbb{G}$. Let Ω_ρ denote the number of probe samples or sequences in the top ρ, then the rank ρ identification rate $R_\rho = \Omega_\rho / J$, the fraction of probes in the top ρ. The rank 1 identification rate is also commonly referred to as the Correct Recognition Rate (CRR).

Similarity between two feature vectors: The similarity score $S(\mathcal{P}_j, \mathcal{G}_m)$ between a probe sample $\mathcal{P}_j \in \mathbb{P}$ and a gallery sample $\mathcal{G}_m \in \mathbb{G}$ is calculated through measuring the distance between the respective feature vectors $\mathbf{p}_j$ and $\mathbf{g}_m$. In this dissertation, seven distance measures are adapted from [95] and studied: the L_1 distance (L1), the L_2 distance (L2), the angle between feature vectors (Angle), the modified Mahalanobis distance (MMD), the modified L_1 distance (ML1), the modified L_2 distance (ML2), and the modified angle distance (MAD), as listed in Table 3.1, where $\mathbf{w}$ is a weight vector and H is the vector length. The first four distance measures are commonly used for measuring vector distances and the last three measures can be viewed as the weighted versions of the first three measures. The similarity score between $\mathbf{p}_j$ and $\mathbf{g}_m$ is obtained as

$$S(\mathcal{P}_j, \mathcal{G}_m) = S(\mathbf{p}_j, \mathbf{g}_m) = -d(\mathbf{p}_j, \mathbf{g}_m), \tag{3.1}$$

using one of the distance measures in Table 3.1. This results in the nearest neighbor classifier in effect. Such a simple classifier is preferred in this dissertation since the focus is on studying the performance mainly contributed by the feature extraction algorithm developed here rather than the classifier. The classification accuracy of the algorithms studied in this dissertation is expected to improve if a more sophisticated classifier such as the support vector machine (SVM) is used instead of the nearest neighbor classifier.

Table 3.1: Seven distance measures for similarity calculation between feature vectors.

Distance	L1	L2	Angle	MMD
$d(\mathbf{a},\mathbf{b})$	$\sum_{h=1}^{H} \lvert\mathbf{a}(h)-\mathbf{b}(h)\rvert$	$\sqrt{\sum_{h=1}^{H} [\mathbf{a}(h)-\mathbf{b}(h)]^2}$	$\frac{-\sum_{h=1}^{H}\mathbf{a}(h)\cdot\mathbf{b}(h)}{\sqrt{\sum_{h=1}^{H}\mathbf{a}(h)^2\sum_{h=1}^{H}\mathbf{b}(h)^2}}$	$-\sum_{h=1}^{H}\frac{\mathbf{a}(h)\cdot\mathbf{b}(h)}{\mathbf{w}(h)}$
Distance	ML1	ML2	MAD	
$d(\mathbf{a},\mathbf{b})$	$\sum_{h=1}^{H}\frac{\lvert\mathbf{a}(h)-\mathbf{b}(h)\rvert}{\mathbf{w}(h)}$	$\frac{\sqrt{\sum_{h=1}^{H}[\mathbf{a}(h)-\mathbf{b}(h)]^2}}{\mathbf{w}(h)}$	$\frac{-\sum_{h=1}^{H}\mathbf{a}(h)\cdot\mathbf{b}(h)}{\mathbf{w}(h)\sqrt{\sum_{h=1}^{H}\mathbf{a}(h)^2\sum_{h=1}^{H}\mathbf{b}(h)^2}}$	

Similarity between two sequences of feature vectors: In gait recognition, a probe gait sequence is often matched against the gallery sequences. A probe sequence $\mathcal{P}_j \in \mathbb{P}$ has N_j samples with corresponding feature vectors: $\{\mathbf{p}_{n_j}, n_j = 1, ..., N_j\}$. A gallery sequence $\mathcal{G}_m \in \mathbb{G}$ has N_m samples with corresponding feature vectors $\{\mathbf{g}_{n_m}, n_m = 1, ..., N_m\}$. To obtain the similarity score $S(\mathcal{P}_j, \mathcal{G}_m)$ between $\mathcal{P}_j$ and $\mathcal{G}_m$, the approach in [6] is adopted, which proposed that the distance calculation process should be symmetric with respect to probe and gallery sequences. If the probe and gallery sequences were interchanged, the computed distance would be identical. The details are described as follows: each probe sample feature $\mathbf{p}_{n_j}$ is matched against the gallery sequence $\mathcal{G}_m$ to obtain

$$S(\mathbf{p}_{n_j}, \mathcal{G}_m) = -\min_{n_m} d(\mathbf{p}_{n_j}, \mathbf{g}_{n_m}) \tag{3.2}$$

and each gallery sample feature $\mathbf{g}_{n_m}$ is matched against the probe sequence $\mathcal{P}_j$ to obtain

$$S(\mathbf{g}_{n_m}, \mathcal{P}_j) = -\min_{n_j} d(\mathbf{g}_{n_m}, \mathbf{p}_{n_j}). \tag{3.3}$$

The similarity score between the probe sequence $\mathcal{P}_j$ and the gallery sequence $\mathcal{G}_m$ is the sum of the mean matching score of $\mathcal{P}_j$ against $\mathcal{G}_m$ and that of $\mathcal{G}_m$ against $\mathcal{P}_j$:

$$S(\mathcal{P}_j, \mathcal{G}_m) = \frac{1}{N_j} \sum_{n_j=1}^{N_j} S(\mathbf{p}_{n_j}, \mathcal{G}_m) + \frac{1}{N_m} \sum_{n_m=1}^{N_m} S(\mathbf{g}_{n_m}, \mathcal{P}_j). \tag{3.4}$$

3.2 The Face Databases

The two widely used public face databases chosen are the Pose, Illumination, and Expression (PIE) database from the Carnegie Mellon University (CMU) [118], and the Facial Recognition Technology (FERET) database [102].

3.2.1 The PIE database

Visually perceived human faces are significantly affected by three factors: the pose, which is the angle they are viewed from, the illumination/lighting condition, and the facial expression such as happy, sad, and anger. The collection of the PIE database is motivated by a need for a database with a fairly large number of subjects imaged a large number of times to cover these three significant factors, i.e., from a variety of different poses, under a wide range of illumination variation, and with several expressions [118].

This database was collected between October 2000 and December 2000 using the CMU 3D Room and it contains 41,368 face images from 68 individuals, with a total size of about 40GB data. The captured images have a size of 640×486. Face images with 13 different poses are captured using 13 synchronized cameras. For the illumination variation, the 3D Room is augmented with a flash system having 21 flashes. Images are captured with and without background lighting, resulting in $21 \times 2 + 1 = 43$ different illumination conditions. In addition, the subjects were asked to pose with four different expressions.

The PIE database can be used for a variety of purposes, including evaluating the

robustness of face recognition systems against the three variations and three-dimensional modeling. In particular, this database has a very large number (around 600 on average) of facial images available for each subject, allowing us to study the effects of the number of training samples (per subject) on the recognition performance. In practice, a subset is usually selected with a specific range of pose, illumination, and expression for experiments so that data sets with various degrees of difficulty can be obtained. A wider range of the three variations leads to a more difficult recognition task.

3.2.2 The FERET database

The FERET database is a widely used database for face recognition performance evaluation. It was constructed through the FERET program, which aims to develop automatic face recognition systems to assist security, intelligence, and law enforcement personnel in the performance of their duties [102]. The face images in this database cover a wide range of variations in pose (viewpoint), illumination, facial expression, acquisition time, ethnicity, and age.

The FERET database was collected in 15 sessions between August 1993 and July 1996, and it contains a total of 14,126 images from 1,199 individuals with views ranging from frontal to left and right profiles. The face images were collected under relatively unconstrained conditions. The same physical setup and location was used in each session to maintain a degree of consistency throughout the database. However, since the equipment was reassembled for each session, images collected on different dates have some minor variation. Sometimes, a second set of images of an individual was captured on a later date, resulting in variations in scale, pose, expression, and illumination of the face. Furthermore, for some people, over two years elapsed between their first and last capturing in order to study changes in a subject's facial appearance over a year.

In this dissertation, the latest color FERET database is used. The images have size of 786×512 and they are encoded with 24 bits. The total data size is around 8GB.

This database has a large number of subjects and it becomes the de facto standard for evaluating face recognition technologies [69], especially in the small sample size scenario, where a smaller number of training samples per subject and a larger number of total subjects lead to a more difficult recognition task [93].

3.2.3 Preprocessing of face images for recognition

In this research, only gray-level facial images are considered without taking color information into account. Moreover, since the focus of this dissertation is on the recognition of faces rather than their detection, all face images from the PIE and FERET databases are manually aligned with manually annotated coordinate information of eyes, cropped, and normalized. There are 20,941 and 5,177 images with the eye coordinate information in the PIE and FERET databases, respectively. These two subsets are first extracted to be the largest evaluation sets for the two databases. The common practice is then followed, where portions of the databases are used for specific studies. The detailed preprocessing procedures are described below and illustrated in Fig. 3.1.

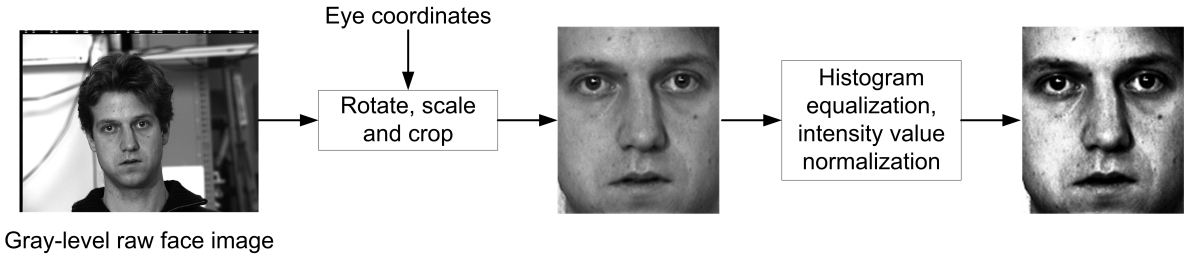


Figure 3.1: Illustration of face image preprocessing.

First, all color images are transformed to gray-level images by taking the luminance component in the YC_bC_r color space. Then, all face images are rotated and scaled so that the centers of the eyes are placed on specific pixels. Next, the image is cropped and normalized to a standard size, followed by histogram equalization, and image intensity values are normalized to have zero mean and unit standard deviation. Finally, each image is represented with 256 gray levels (eight bits) per pixel, and naturally as a second-order

tensor (Fig. 1.2(a), page 7). Figure 3.2 shows 160 near-frontal face images for one subject in the PIE database, and Fig. 3.3 shows some sample face images from two subjects in the FERET database.

Figure 3.2: Sample face images of one subject from the CMU PIE database.

Figure 3.3: Examples of face images from two subjects in the FERET database.

3.3 The Gait Database

In this dissertation, the HumanID Gait Challenge data sets version 1.7 (V.1.7) from the University of South Florida (USF) [109] is used for evaluating the performance on gait recognition. This database captures the variations of a number of covariates for a large group of people. It has emerged as a standard testbed for new gait recognition algorithms [13]. Other databases are limited in size, variations, capturing conditions, or

of high resolutions [109,98]. This section describes this USF database and then introduces how to obtain gait samples from a gait silhouette sequence.

3.3.1 The USF Gait Challenge database

As an emerging technology in its infancy, gait recognition has many open questions to answer. It is important to investigate the conditions under which this problem is "solvable", and to find out what factors affect gait recognition and to what extent. The HumanID Gait Challenge Problem is thus introduced by the USF in order to assess the potential of gait recognition by providing a means for measuring progress and characterizing the properties [109].

This challenge problem consists of a baseline algorithm, a large data set, and a set of 12 experiments. The baseline algorithm extracts silhouettes through background subtraction and performs recognition via temporal correlation of silhouettes. The data was collected outdoors since gait, as a biometric, is most appropriate in outdoor at-a-distance settings where other biometrics are difficult to capture [109]. Figure 3.4 shows two sample frames from this database. The 12 experiments, in increasing difficulty, examine the effects of five covariates on recognition performance: change in viewing angle, change in shoe type, change in walking surface, carrying or not carrying a briefcase, and temporal (time) differences [109], where the time covariate implicitly includes other changes naturally occur between video acquisition sessions such as change of shoes and cloths, change in the outdoor lighting conditions, and inherent variation in gait over time. These covariates either affect gait or affect the extraction of gait features from images. They are selected, based on logistical issues and collection feasibility, from a list of factors compiled through the discussions with researchers at CMU, Maryland, MIT, Southampton, and Georgia Tech about potentially important covariates for gait analysis. It is shown in [109] that the shoe type has the least impact on the performance, next is the viewpoint, the third is briefcase, then surface type (flat concrete surface and typical grass lawn surface),

Figure 3.4: Sample frames from the Gait Challenge data sets.

and time (six months) difference has the greatest impact. The latter two are the most "difficult" covariates to deal with. In particular, it was found that the surface covariate impacts the gait period more than other covariates. Since its release, this database has made significant contributions to the advancement of the gait recognition technology.

This dissertation evaluates gait recognition performance on the USF gait database V.1.7, which was collected in May 2001 and widely used in the gait recognition community [82, 72, 73, 65]. This database consists of 452 sequences from 74 subjects walking in elliptical paths in front of the camera. The raw video frames are of size 720×480 in 24-bit RGB and a subject's size in the back portion of the ellipse is on average 100 pixels in height. The total data is around 300GB. For each subject, there are three covariates: viewpoint (left or right), shoe type (two different types, A or B), and surface type (grass or concrete). The gallery set contains 71 sequences (subjects) and seven experiments (probe sets) are designed for human identification as shown in Table 3.2. The capturing condition for each probe set is summarized in the parentheses after the probe name in the Table, where C, G, A, B, L, and R stand for concrete surface, grass surface, shoe type A, shoe type B, left view, and right view, respectively. For instance, the capturing condition of the gallery set is GAR (Grass surface, shoe type A and Right view). Each set has only one sequence for a subject. Subjects are unique in the gallery and each probe set. There are no common sequences between the gallery set and any of the probe

sets. In addition, all the probe sets are distinct.

Table 3.2: The characteristics of the gait data from the USF Gait Challenge data sets version 1.7.

Gait data set	Number of sequences (samples)	Difference from the gallery
Gallery (GAR)	71 (731)	-
Probe A(GAL)	71 (727)	View
Probe B(GBR)	41 (423)	Shoe
Probe C(GBL)	41 (420)	Shoe, view
Probe D(CAR)	70 (682)	Surface
Probe E(CBR)	44 (435)	Surface, shoe
Probe F(CAL)	70 (685)	Surface, view
Probe G(CBL)	44 (424)	Surface, shoe, view

Figure 3.5 illustrates the process of silhouette extraction through background subtraction, where a background model is estimated from the input raw gait sequences and then it is subtracted to get the silhouettes. The extracted silhouettes are then cropped and resized to a standard size. The silhouettes data extracted through the baseline algorithm is provided by the USF and it is widely used in the gait recognition literature [82, 5, 6, 4]. Thus, this silhouettes data is used for the gait recognition experiments in this dissertation. Although an automatic silhouette extraction algorithm [76] is developed in this research, it is not included here since the focus of this dissertation is on recognition rather than silhouette extraction.

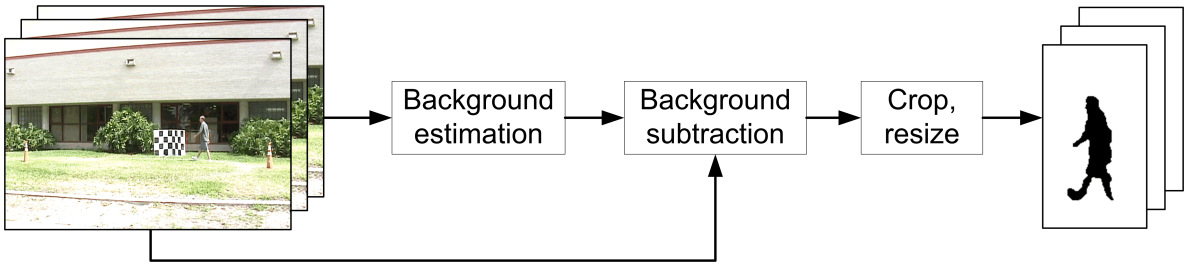


Figure 3.5: Illustration of the silhouette extraction process.

3.3.2 Normalization of tensorial gait samples

While in many recognition problems, an input sample is unambiguously defined, such as iris, face or fingerprint images, there is no obvious definition of a gait sample. This dissertation proposes to treat each half gait cycle as a data sample. Thus, a gait sample is a third-order tensor, and the spatial column space, row space, and the time space account for its three modes, as shown in Fig. 1.2(b) (page 7).

To obtain half cycles, a gait silhouette sequence is partitioned in a way similar to that used in [109]. The number of foreground pixels is counted in the bottom half of each silhouette since legs are the major visible moving body parts from a distance. This number will reach a maximum when the two legs are farthest apart and drop to a minimum when the legs overlap. The sequence of these numbers is smoothed with a running average filter and the minimums in this number sequence partition the sequence into several half gait cycles. Following the proposal above, there are 731 gait samples in the gallery set and each subject has an average of roughly ten samples available. The number of samples for each set is indicated in the parentheses following the number of sequences in Table 3.2. The proposed simple partition method may be improved further by taking the periodic nature of the gait cycles into account, such as the more robust cycle partitioning algorithm in [49], while this is not investigated in this dissertation.

Each frame of the gait silhouette sequences from the USF data sets is of standard size 128×88, but the number of frames in each gait sample obtained through half cycle partition has some variation. Before feeding the gait samples to a subspace learning algorithm, the tensorial inputs need to be normalized to the same dimension in each mode. Since the row and column dimensions are normalized by default, only the time mode, i.e., the number of frames in each gait sample, is subject to normalization. The normalized time mode dimension is chosen to be 20, roughly the average number of frames in each gait sample. Thus, each gait sample has a canonical representation of $I_1 \times I_2 \times I_3 = 128 \times 88 \times 20$. In the following, a simple procedure for this time-mode

normalization is described.

Consider one gait sample of size $I_1 \times I_2 \times D_3$. While there are sophisticated algorithms available, such as mapping a gait cycle to a unit circle using nonlinear interpolation [64], conventional interpolation algorithms, such as linear interpolation, can be applied to the time-mode normalization as well. Hence, in this research, each 3-mode (time-mode) vector is interpolated linearly from the original size $D_3 \times 1$ to the normal size $I_3 \times 1$, followed by binarization to get binary silhouettes. Figure 3.6 shows three gait samples obtained this way from the USF gait database V.1.7 by concatenating the frames on a row.

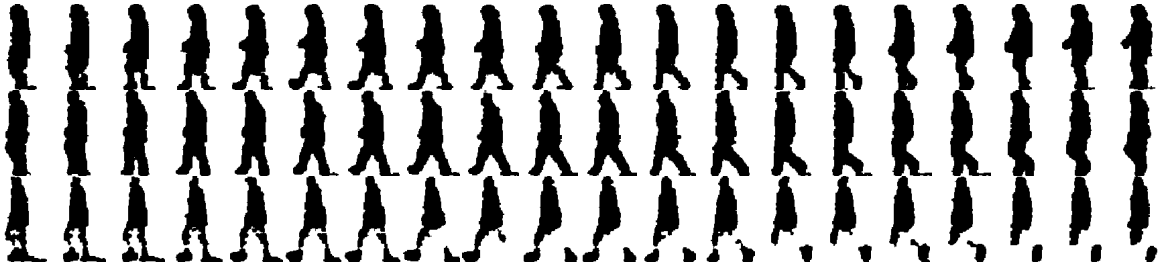


Figure 3.6: Three gait samples from the USF gait database V.1.7, shown by concatenating frames in rows.

3.4 Review on Multilinear Subspace Learning Algorithms

As mentioned in Sec. 1.4.2 (page 7), the application of linear subspace learning algorithms such as PCA or LDA to the recognition of face or gait objects requires their reshaping into vectors in a very-high-dimensional space. This results in the estimation of a large number of parameters, and also high computational and memory demands. For example, vectorizing a gait sample of size $128 \times 88 \times 20$ results in a $225,280 \times 1$ vector, the singular value decomposition (SVD) or eigen-decomposition processing of which may be beyond the processing capabilities of many computing devices. Beyond implementation issues,

reshaping breaks the natural structure and correlation, and removes redundancies and dependencies in their natural tensorial forms, from which more compact or useful representations may be obtained [156, 155]. Based on this motivation, multilinear subspace learning algorithms [156, 146, 150, 124] operating directly on the tensorial representations rather than their vectorial versions are emerging, partly due to the recent developments in [59, 60, 2]. This section reviews these new developments. Due to the fundamentality and importance of PCA and LDA, the focus is on the multilinear extensions of these two classical linear algorithms. Figures 3.7(a) and 3.7(b) provide an overview of these existing unsupervised and supervised multilinear subspace learning algorithms under the multilinear subspace learning framework introduced in Chapter 2, respectively, and they are discussed in the following.

3.4.1 Unsupervised multilinear subspace learning through tensor-to-tensor projection

The development of unsupervised multilinear subspace learning started with the treatment of images directly as matrices rather than vectors.

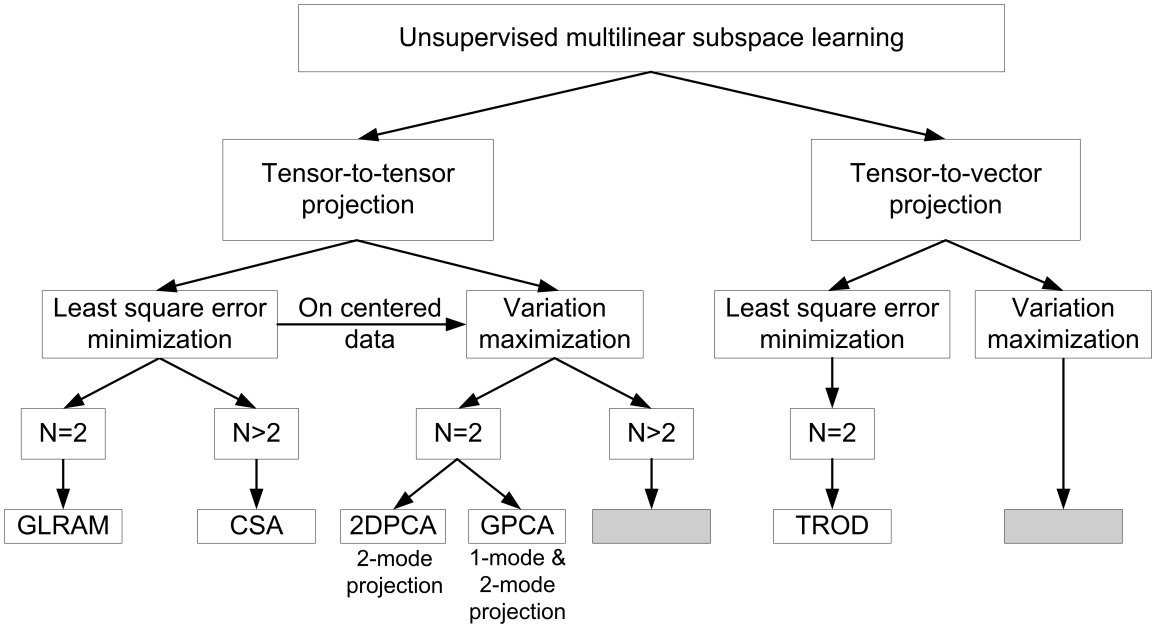
A two-dimensional PCA (2DPCA) algorithm is proposed in [153]. This algorithm solves for a linear transformation $\mathbf{U} \in \mathbb{R}^{I_2 \times P_2}$ $(P_2 < I_2)$ that projects an image $\mathbf{X}_m \in \mathbb{R}^{I_1 \times I_2}$ to

$$\mathbf{Y}_m = \mathbf{X}_m \mathbf{U} = \mathbf{X}_m \times_2 \mathbf{U}^T \in \mathbb{R}^{I_1 \times P_2} \tag{3.5}$$

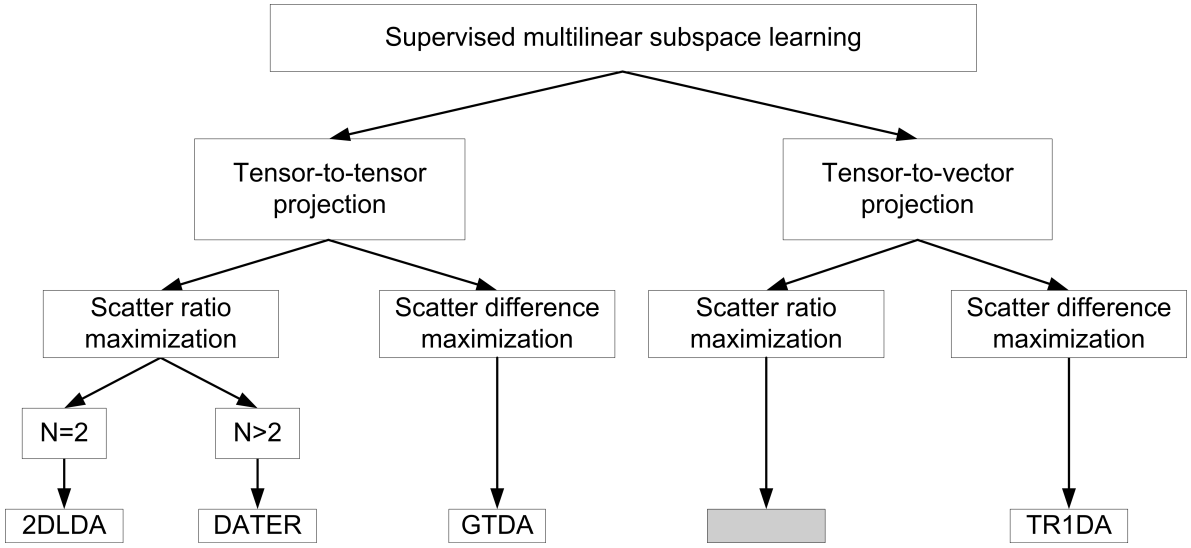
while maximizing the variance measure

$$\sum_{m=1}^{M} \parallel \mathbf{Y}_m - \bar{\mathbf{Y}} \parallel_F^2 = \sum_{m=1}^{M} trace\left(\mathbf{U}^T (\mathbf{X}_m - \bar{\mathbf{X}})^T (\mathbf{X}_m - \bar{\mathbf{X}}) \mathbf{U}\right), \tag{3.6}$$

where $\bar{\mathbf{X}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{X}_m$ and $\bar{\mathbf{Y}} = \frac{1}{M} \sum_{m=1}^{M} \mathbf{Y}_m$. This algorithm works directly on image matrices (second-order tensors) but there is only one linear transformation of the 2-mode.



(a)



(b)

Figure 3.7: Overview of existing (a) unsupervised multilinear subspace learning algorithms, and (b) supervised multilinear subspace learning algorithms. The shaded empty boxes indicate the approaches that have not been studied.

Thus, the image data is projected in the 2-mode (the row mode) only while the projection in the 1-mode (the column mode) is ignored (or effectively an identity transformation), resulting in poor dimensionality reduction.

A more general algorithm named the generalized low rank approximation of matrices

(GLRAM) was introduced in [155], which takes into account the spatial correlation of the image pixels within a localized neighborhood and applies two linear transforms to both the left and right sides of input image matrices. This algorithm solves for two linear transformations $\mathbf{U}^{(1)} \in \mathbb{R}^{I_1 \times P_1}$ $(P_1 < I_1)$ and $\mathbf{U}^{(2)} \in \mathbb{R}^{I_2 \times P_2}$ $(P_2 < I_2)$ that project an image $\mathbf{X}_m \in \mathbb{R}^{I_1 \times I_2}$ to

$$\mathbf{Y}_m = \mathbf{U}^{(1)^T} \mathbf{X}_m \mathbf{U}^{(2)} = \mathbf{X}_m \times_1 \mathbf{U}^{(1)^T} \times_2 \mathbf{U}^{(2)^T} \in \mathbb{R}^{P_1 \times P_2} \tag{3.7}$$

while minimizing the least-square (reconstruction) error measure

$$\sum_{m=1}^{M} \parallel \mathbf{X}_m - \mathbf{U}^{(1)} \mathbf{Y}_m \mathbf{U}^{(2)^T} \parallel_F^2 = \sum_{m=1}^{M} \parallel \mathbf{X}_m - \mathbf{Y}_m \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \parallel_F^2 . \tag{3.8}$$

Thus, projections in both modes are involved and better dimensionality reduction results than [153] are obtained according to [155].

Although GLRAM exploits both modes for subspace learning, it is formulated for matrices (second-order tensors) only. Recently, the so-called concurrent subspaces analysis (CSA) is formulated in [146] for general tensor objects, which can be considered as a further generalization of GLRAM for higher-order tensors. The solution is built in a manner similar to the best Rank-$(R_1, R_2, ..., R_N)$ approximation in [60]. This algorithm solves for more general multilinear transformations $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, P_n \leq I_n, n = 1, ..., N\}$ that project a tensor $\mathcal{X}_m \in \mathbb{R}^{I_1 \times ... \times I_N}$ to

$$\mathcal{Y}_m = \mathcal{X}_m \times_1 \mathbf{U}^{(1)^T} \times_2 ... \times_N \mathbf{U}^{(N)^T} \in \mathbb{R}^{P_1 \times ... \times P_N} \tag{3.9}$$

while minimizing the following reconstruction error metric

$$\sum_{m=1}^{M} \parallel \mathcal{X}_m - \mathcal{Y}_m \times_1 \mathbf{U}^{(1)} \times_2 ... \times_N \mathbf{U}^{(N)} \parallel_F^2 . \tag{3.10}$$

Unfortunately, CSA appears to be sensitive to parameter settings, as shown in [146]. Furthermore, there is no systematic way to determine the tensor subspace dimensionality. As the number of possible subspace dimensions for most tensor objects is extremely high (e.g., there are $225,280$ possible subspace dimensions for the gait recognition problem discussed in Section 3.3), exhaustive testing for determination of subspace dimension is not feasible. Consequently, the algorithmic solution of [146] can not be used to effectively determine subspace dimensionality in a comprehensive and systematic manner.

Whereas GLRAM and CSA advanced the unsupervised multilinear subspace learning, they are both formulated with the objective of optimal reconstruction or approximation of tensors. Therefore, they ignored an important centering step in unsupervised subspace learning algorithms developed for recognition, such as the classical PCA, where the data is centered first before obtaining the subspace projection. It should be pointed out that for the reconstruction or approximation problem, centering is not essential, as the (sample) mean is the main focus of attention. However, in recognition applications where the solutions involve eigenproblems, non-centering (in other words, an average different from zero) can potentially affect the eigen-decomposition in each mode and lead to a solution that captures the variation with respect to the origin rather than capturing the true variation of the data (with respect to the data center). This will be illustrated in Chapter 7.

In contrast, the generalized PCA (GPCA) proposed in [156] is an extension of PCA that works on matrices. GPCA is exactly the same as GLRAM except that the projection takes the centered data $\tilde{\mathbf{X}}_m = \mathbf{X}_m - \bar{\mathbf{X}}$ rather than the original coordinate $\mathbf{X}_m$ as input. Nonetheless, this work is formulated only for matrices, and important issues such as initialization and subspace dimensionality determination are not studied either. Moreover, the effect of centering on recognition problem is not investigated.

3.4.2 Unsupervised multilinear subspace learning through tensor-to-vector projection

There is only one existing algorithm of unsupervised multilinear subspace learning through tensor-to-vector projection. It is the tensor rank-one decomposition (TROD) algorithm introduced in 2001, formulated only for image matrices [116]. This algorithm looks for a second-order tensor-to-vector projection $\{\mathbf{u}_p^{(1)^T}, \mathbf{u}_p^{(2)^T}\}_{p=1}^{P}$ that project an image $\mathbf{X}_m \in \mathbb{R}^{I_1 \times I_2}$ to

$$\mathbf{y}_m = \mathbf{X}_m \times_{n=1}^{2} \{\mathbf{u}_p^{(n)^T}, n = 1, 2\}_{p=1}^{P}, \in \mathbb{R}^{P \times 1} \tag{3.11}$$

while minimizing the following least-square (reconstruction) error measure

$$\sum_{m=1}^{M} \left|\left| \mathbf{X}_m - \sum_{p=1}^{P} \mathbf{y}_m(p) \cdot \mathbf{u}_p^{(1)} \mathbf{u}_p^{(2)^T} \right|\right|_F^2 \tag{3.12}$$

to obtain $\{\mathbf{u}_p^{(1)^T}, \mathbf{u}_p^{(2)^T}\}$. Thus, there are P steps, with each solving one elementary projection. The solution of TROD relies on a heuristic procedure of successive residue calculation, i.e., after obtaining the pth elementary multilinear projection $\{\mathbf{u}_p^{(1)^T}, \mathbf{u}_p^{(2)^T}\}$, the input image is replaced by its residue as

$$\mathbf{X}_m = \mathbf{X}_m - \mathbf{y}_m(p) \cdot \mathbf{u}_p^{(1)} \mathbf{u}_p^{(2)^T}. \tag{3.13}$$

Though this algorithm is the first and only existing work in this category, it has many limitations mentioned in Sec. 3.4.1 too. It is formulated only for matrices and the input data is not centered either.

In addition, none of these existing unsupervised multilinear subspace learning algorithms takes into account the correlations among features and shares an important property with PCA, i.e., zero-correlation among extracted features. It is well-known that PCA derives uncorrelated features, which contain minimum redundancy and ensure

linear independence among features. Uncorrelated features can also greatly simplify the subsequent classification task and they are highly desirable in recognition applications. Instead, most of existing unsupervised multilinear subspace learning algorithms produce orthogonal bases in each mode. Although uncorrelated features imply orthogonal projection bases in PCA, this is not necessarily true for its multilinear extension.

3.4.3 Supervised multilinear subspace learning through tensor-to-tensor projection

Besides the unsupervised multilinear subspace learning algorithms reviewed above, there are also supervised multilinear subspace learning algorithms proposed in the literature, where the class labels are used in the learning process.

Like GLRAM and GPCA, the two-dimensional LDA (2DLDA) introduced in 2004 [157] solves for two linear transformations $\mathbf{U}^{(1)} \in \mathbb{R}^{I_1 \times P_1}$ $(P_1 < I_1)$ and $\mathbf{U}^{(2)} \in \mathbb{R}^{I_2 \times P_2}$ $(P_2 < I_2)$ that project an image $\mathbf{X}_m \in \mathbb{R}^{I_1 \times I_2}$ to $\mathbf{Y}_m$ as in (3.16), but with a different objective criterion. For M image samples $\{\mathbf{A}_m\}$, the between-class and within-class scatter measures are defined as

$$\Psi_{B_{\mathbf{A}}} = \sum_{c=1}^{C} N_c \parallel \bar{\mathbf{A}}_c - \bar{\mathbf{A}} \parallel_F^2 = trace\left(\sum_{c=1}^{C} N_c \cdot \left(\bar{\mathbf{A}}_c - \bar{\mathbf{A}}\right)\left(\bar{\mathbf{A}}_c - \bar{\mathbf{A}}\right)^T\right) \tag{3.14}$$

and

$$\Psi_{W_{\mathbf{A}}} = \sum_{m=1}^{M} \parallel \mathbf{A}_m - \bar{\mathbf{A}}_{c_m} \parallel_F^2 = trace\left(\sum_{m=1}^{M} \left(\mathbf{A}_m - \bar{\mathbf{A}}_{c_m}\right)\left(\mathbf{A}_m - \bar{\mathbf{A}}_{c_m}\right)^T\right), \tag{3.15}$$

respectively. In these definitions, the mean image is $\bar{\mathbf{A}} = \frac{1}{M}\sum_m \mathbf{A}_m$ and the class mean image is $\bar{\mathbf{A}}_c = \frac{1}{N_c}\sum_{m,c_m=c} \mathbf{A}_m$. The image-based discrimination criterion is then defined as the scatter ratio $\Psi_{B_{\mathbf{Y}}}/\Psi_{W_{\mathbf{Y}}}$.

Later, a more general extension, the discriminant analysis with tensor representa-

tion (DATER)[1] was proposed to perform discriminant analysis on more general tensorial inputs [150]. Like CSA, the DATER algorithm solves for more general multilinear transformations $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, P_n \leq I_n, n = 1, ..., N\}$ that project a tensor $\mathcal{X}_m \in \mathbb{R}^{I_1 \times ... \times I_N}$ to $\mathcal{Y}_m$ as in (3.9). The tensor-based discrimination objective criterion is formulated based on Definition 2.2 (page 31) and the tensor-based scatter ratio $\Psi_{B_\mathcal{Y}}/\Psi_{W_\mathcal{Y}}$ is maximized. However, this algorithm does not converge and it appears to be sensitive to parameter settings, as shown in [145] and also Fig. 3.8 where the evolution of the objective criterion is plotted against the iteration number for a training on gait samples. Furthermore, the work in [145] provides no systematic way to determine the tensor subspace dimensionality either. As mentioned in the discussion of CSA in Sec. 3.4.1, the exhaustive testing method in [145] is not practical in determining the subspace dimensionality.

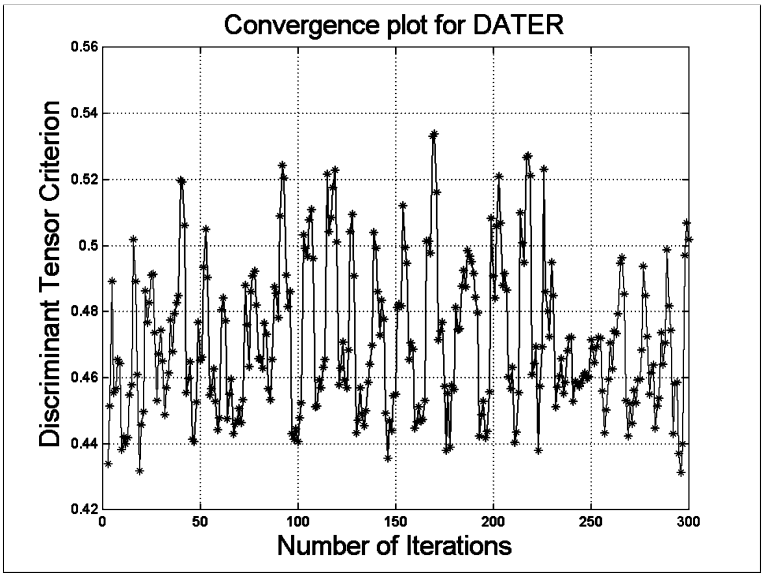


Figure 3.8: The evolution of the objective criterion over iterations when the DATER algorithm in [152] is applied on tensorial gait samples.

In [124], the general tensor discriminant analysis (GTDA) algorithm is proposed. The GTDA algorithm also solves for multilinear transformations $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times P_n}, P_n \leq I_n, n = 1, ..., N\}$ that project a tensor $\mathcal{X}_m \in \mathbb{R}^{I_1 \times ... \times I_N}$ to $\mathcal{Y}_m$ as in CSA and DATER. The

[1]Here, the name used when the algorithm was first proposed is adopted as it is more commonly referred to in the literature.

difference with DATER is that it maximizes a tensor-based scatter difference criterion ($\Psi_{B_\mathcal{Y}} - \zeta\Psi_{W_\mathcal{Y}}$), where ζ is a tuning parameter [70]. Although this algorithm is shown to have good convergence property [124], the criterion used is dependent on the coordinate system, as pointed out in [30], and the heuristic determination of the tuning parameter ζ in [124] is not guaranteed to be optimal. Hence, in [124], this algorithm is only used as a preprocessing tool.

3.4.4 Supervised multilinear subspace learning through tensor-to-vector projection

As in the unsupervised case, there is only one existing algorithm of supervised multilinear subspace learning through the tensor-to-vector projection. It is the tensor rank-one discriminant analysis (TR1DA) algorithm proposed in [142,123], derived from the TROD algorithm [116]. The TR1DA algorithm is formulated for general tensor objects and it looks for a tensor-to-vector projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ that project a tensor $\mathcal{X}_m \in \mathbb{R}^{I_1 \times ... \times I_N}$ to

$$\mathbf{y}_m = \mathcal{X}_m \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}, \in \mathbb{R}^{P \times 1} \tag{3.16}$$

while maximizing the scalar scatter difference criterion ($S_{B_p}^{\mathbf{y}} - \zeta \cdot S_{W_p}^{\mathbf{y}}$) to obtain $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$. As in TROD, there are P steps as well. This criterion is formulated based on Definition 2.5 (page 32) and as in GTDA, ζ is a tuning parameter. Therefore, the criterion is also dependent on the coordinate system and there is no way to determine the optimal ζ either. Furthermore, this algorithm also relies on the repeatedly-calculated residues in (3.13), originally proposed in [53] for tensor approximation. The adoption of this heuristic procedure here lacks theoretical explanation for a discriminative criterion.

Moreover, in these existing supervised multilinear subspace learning algorithms, the attention focused mainly on the objective criterion in terms of (either the ratio of or the

difference between) the between-class scatter and the within-class scatter since it is well-known that the classical LDA aims to maximize the Fisher's discrimination criterion. However, they did not take the correlations among features into account, as in the existing unsupervised multilinear subspace learning algorithms. In other words, an important property of the classical LDA is ignored in these developments: the classical LDA derives uncorrelated features, as proved in [46, 158], where the uncorrelated LDA (ULDA) introduced in [45] is shown to be equivalent to the classical LDA. As mentioned in Sec. 3.4.2, uncorrelated features contain minimum redundancy and ensure independence of features so they are highly desirable in many applications [158].

Further, as mentioned in Sec. 1.4.2 (page 7), although the small sample size problem is reduced in multilinear subspace learning, the number of parameters to be estimated in supervised multilinear subspace learning still far exceeds the number of samples available for their accurate estimation in most practical situations. Nevertheless, there is no attempt in existing supervised multilinear subspace learning algorithms to tackle the small sample size problem in the multilinear case.

3.4.5 Related prior multilinear algorithms

Multilinear algebra, the extension of linear algebra, has been well studied in mathematics around the middle of the 20th century [33, 57]. It builds on the concept of tensors and develops the theory of tensor spaces.

A popular early application of multilinear algebra is the so-called multi-way analysis, developed in psychometrics and chemometrics for factor analysis of multi-way data sets[2] [127, 12, 23, 35, 56, 55], starting from the 60s and 70s. There are two main types of decomposition methods developed in this field: the Tucker decomposition [127, 2, 59], and the canonical decomposition (CANDECOMP) [12, 2, 59], which is also known as the

[2]Multi-way (multivariate) data sets are higher-order tensors characterized by several sets of categorical variables that are measured in a crossed fashion [23, 55].

parallel factors (PARAFAC) decomposition [35, 2, 59].

In the 90s, the developments in the field of higher-order statistics of multivariate stochastic variables have attracted interests in higher-order tensors from the signal processing community [16, 15, 58]. The Tucker decomposition was reintroduced and further developed in [59] as the higher-order singular value decomposition (HOSVD) solution, an extension of the SVD to higher-order tensors. Its computation leads to the calculation of N different matrix SVDs of unfolded matrices. The ALS algorithm for the best Rank-$(R_1, R_2, ..., R_N)$ approximation of higher-order tensors was studied in [60], where tensor data was projected into a lower dimensional tensor space iteratively. The application of the HOSVD truncation and the best Rank-$(R_1, R_2, ..., R_N)$ approximation to dimensionality reduction in independent component analysis (ICA) was discussed in [61].

The development in [59,60] has led to the development of new multilinear algorithms and the exploration of new application areas. Multilinear analysis of biometric data is pioneered by the TensorFace method [131, 132, 134, 133], which employs the multilinear algorithms proposed in [59, 60] to analyze the factors involved in the formation of facial images. Similar analysis has also been done for motion signatures [130] and gait sequences [64]. However, in these multiple factor analysis work, input data such as images or video sequences are still represented as vectors and these vectors are arranged into a tensor according to the multiple factors involved in their formation for subsequent analysis. Such tensor formation needs a large number of training samples captured under various conditions, which is often impractical and may have the missing-data problem. Furthermore, the tensor data size is usually huge, leading to high memory and computational demands.

Finally, besides the multilinear extensions of the linear subspace learning algorithms, the multilinear extensions of linear graph-embedding algorithms were also introduced in [37, 20, 151, 144, 41], in a similar fashion as the existing multilinear subspace learning algorithms reviewed in this chapter.

3.5 Summary

This chapter has provided the background materials for this dissertation. First, the issues regarding recognition performance evaluation are discussed. Then, this chapter has described the three databases to be used in the performance evaluation of the multilinear subspace learning algorithms. Finally, existing literatures related to multilinear subspace learning are reviewed and discussed within the multilinear subspace learning framework presented in Chapter 2. While the development of multilinear subspace learning is encouraging, this field is still in its infancy and needs more research work. In particular, the limitations of the existing multilinear subspace learning methodologies have been pointed out, and by referring to Figs. 3.7(a) and 3.7(b), there are unexplored directions in both unsupervised and supervised multilinear subspace learning. The following is a summary of the conclusions drawn from this review:

1. In the approach of the tensor-to-tensor projection, there is no existing unsupervised multilinear subspace learning algorithm for general tensors derived from the perspective of variance maximization, i.e, taking the centering of data into considerations. Although GPCA [156] centers the data, the algorithm is formulated on second-order tensors only and the effect of centering is not studied. Thus, there is a need to develop a more general unsupervised multilinear subspace learning for general tensors based on the tensor-to-tensor projection, as indicated by the shaded empty box in Fig. 3.7(a) under the tensor-to-tensor projection. Furthermore, systematic determination of the subspace dimensionality is an open problem to be addressed, and other design issues of paramount importance in practical applications such as the initialization, termination, and convergence of the algorithm are not systematically investigated in the literature.

2. In unsupervised multilinear subspace learning through the tensor-to-vector projection, there is no existing work taking the approach of variation maximization, a

standard approach of PCA, as indicated by the shaded empty box in Fig. 3.7(a) under the tensor-to-vector projection.

3. It is well-known that PCA derives uncorrelated features. However, none of the existing unsupervised multilinear subspace learning algorithms has this property. Instead, most of them produce orthogonal bases in each mode, which does not lead to uncorrelated features in the multilinear case, unlike in the linear case.

4. In supervised multilinear subspace learning through the tensor-to-vector projection, there is no existing work attempting to maximize the scatter ratio, a classical measure in LDA, as indicated by the only shaded empty box in Fig. 3.7(b).

5. Similar to PCA, LDA also derives uncorrelated features but existing supervised multilinear subspace learning algorithms are not aware of this property while concentrating only on constructing discrimination criteria.

6. Existing supervised multilinear subspace learning algorithms have not made any attempt in addressing the small sample size problem in the multilinear setting.

In order to overcome the above listed limitations and advance the current state of multilinear subspace learning, a systematic treatment on this topic has been given in Chapter 2. Next, several algorithms in this field will be proposed in the following chapters to address the review conclusions above. Detailed analysis, derivations, and comparisons will be presented. The recognition performance of the proposed algorithms will be evaluated on the face and gait databases described in this chapter. The first work to be introduced in the next chapter is the MPCA solution, which attempts to address the first conclusion above drawn from the literature review.

Chapter 4

Multilinear Principal Component Analysis

4.1 Introduction

This chapter introduces MPCA, the first unsupervised multilinear subspace learning algorithm for general tensors targeting at variance maximization as in the classical PCA rather than reconstruction error minimization. MPCA aims to solve for a tensor-to-tensor projection that allows projected tensors to capture most of the variation present in the original tensors. It can be considered as the higher-order extension of GPCA [156], which is formulated only for second-order tensors. Thus, this chapter addresses the shaded empty box in Fig. 3.7(a) (page 51) under the tensor-to-tensor projection.

The solution for MPCA is iterative in nature. It proceeds by decomposing the original problem to a series of multiple projection subproblems. Consequently, design issues of paramount importance in practical applications, such as initialization, projection order, termination, and convergence of the algorithm, are discussed. Methods for systematic determination of the subspace dimensionality are proposed and analyzed. The relationships between MPCA and existing PCA-based solutions are revealed. In addition,

the eigentensors and n-mode eigenvalues are defined as counterparts of the eigenvectors and eigenvalues in the classical PCA. The geometric interpretation of these concepts is provided, enabling a deeper understanding of the MPCA algorithm and facilitating its application.

Furthermore, a discriminative tensor feature selection mechanism is introduced together with a novel weighting method for better recognition. The combination of MPCA and LDA is discussed as well. Moreover, the integration of MPCA with the ensemble-based discriminant learning [93] is investigated for better generalization performance. In this approach, a subset of the features extracted by MPCA is fed into a LDA-style booster, which gives another way of learner weakness control in addition to computational efficiency. The LDA learner in [93] is modified by adopting a simpler weighted pairwise between-class scatter matrix and introducing a regularization term in the within-class scatter matrix so that the complex and nonlinear distribution of patterns is taken into account.

The rest of this chapter is organized as follows. In Section 4.2, the problem of MPCA is formulated and an iterative solution is derived. The connections to the existing PCA-based solutions are then pointed out. Section 4.3 gives detailed discussions on the design issues including the initialization procedures, the projection order, the termination criteria, the convergence, and the determination of the subspace dimensionality. In addition, the computational aspects of the proposed method are also discussed in Section 4.3. The selection of discriminative MPCA features and the combination with LDA are presented in Section 4.4. The combination of MPCA with the boosting technology is introduced in Section 4.5. Section 4.6 constructs three synthetic data sets for experimental study of the MPCA properties. Finally, Section 4.7 summarizes this chapter.

4.2 The MPCA Algorithm

This section defines the MPCA problem first. The solution to this problem is then derived and connections with other PCA-based algorithms are discussed.

4.2.1 The MPCA problem

From Definition 2.1 (page 30), the MPCA problem is defined as follows.

A set of M tensor objects $\{\mathcal{X}_1,\ \mathcal{X}_2,\ ...,\ \mathcal{X}_M\}$ is available for training. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ assumes values in a 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 MPCA objective is to define a multilinear tensor-to-tensor projection $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, ..., N\}$ that maps 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 $P_n \leq I_n$, for $n = 1, ..., N$):

$$\mathcal{Y}_m = \mathcal{X}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, ..., M, \tag{4.1}$$

such that $\{\mathcal{Y}_m \in \mathbb{R}^{P_1} \bigotimes \mathbb{R}^{P_2} ... \bigotimes \mathbb{R}^{P_N}, m = 1, ..., M\}$ captures most of the variation observed in the original tensor objects, assuming that these variation are measured by the total scatter defined for tensors in Definition 2.1 (page 30).

In other words, the MPCA objective is the determination of the N projection matrices $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, 2, ..., N\}$ that maximize the total tensor scatter $\Psi_{\mathcal{Y}}$:

$$\{\tilde{\mathbf{U}}^{(n)}, n = 1, 2, ..., N\} = \arg \max_{\tilde{\mathbf{U}}^{(1)}, \tilde{\mathbf{U}}^{(2)}, ..., \tilde{\mathbf{U}}^{(N)}} \Psi_{\mathcal{Y}}, \tag{4.2}$$

where $\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \parallel \mathcal{Y}_m - \bar{\mathcal{Y}} \parallel_F^2$. Here, the dimensionality P_n for each mode is assumed to be known or pre-determined first. Discussions on the adaptive determination of P_n, when it is not known in advance, will be presented in Section 4.3.6.

4.2.2 The derivation of the MPCA solution

Unfortunately, as in other multilinear subspace learning algorithms, the N projection matrices need the determination of N sets of parameters, and these N sets of parameters are inter-dependant (except for $N = 1$ or $P_n = I_n$ for all n). This results in a highly nonlinear problem with no known optimal solution that allows for the simultaneous optimization of the N projection matrices.

Thus, the alternating projection method in the ALS algorithm [12,35,56] is followed. Since the projection to an Nth-order tensor subspace consists of N projections to N vector subspaces, N optimization subproblems can be solved by finding the $\tilde{\mathbf{U}}^{(n)}$ that maximizes the scatter in the n-mode vector subspace, conditioned on the projection matrices in the other modes. The solution for such a subproblem is given in the following theorem.

Theorem 4.1. *Let* $\{\tilde{\mathbf{U}}^{(n)}, n = 1, ..., N\}$ *be the solution to Equation (4.2). Then, given all the other projection matrices* $\{\tilde{\mathbf{U}}^{(1)}, ..., \tilde{\mathbf{U}}^{(n-1)}, \tilde{\mathbf{U}}^{(n+1)}, ..., \tilde{\mathbf{U}}^{(N)}\}$*, the projection matrix* $\tilde{\mathbf{U}}^{(n)}$ *consists of the* P_n *eigenvectors corresponding to the largest* P_n *eigenvalues of the matrix*

$$\mathbf{\Phi}^{(n)} = \sum_{m=1}^{M} \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^{T} \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right)^{T}, \tag{4.3}$$

where

$$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} = \left(\tilde{\mathbf{U}}^{(n+1)} \otimes \tilde{\mathbf{U}}^{(n+2)} \otimes ... \otimes \tilde{\mathbf{U}}^{(N)} \otimes \tilde{\mathbf{U}}^{(1)} \otimes \tilde{\mathbf{U}}^{(2)} \otimes ... \tilde{\mathbf{U}}^{(n-1)}\right). \tag{4.4}$$

Proof. The proof of Theorem 1 is given in Appendix A.1. □

As seen from the above, the product $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^{T}$ depends on $\{\tilde{\mathbf{U}}^{(1)}, ..., \tilde{\mathbf{U}}^{(n-1)}, \tilde{\mathbf{U}}^{(n+1)}, ..., \tilde{\mathbf{U}}^{(N)}\}$, indicating that the optimization of $\tilde{\mathbf{U}}^{(n)}$ depends on the projections in other modes. Therefore, there is no closed-form solution to this maximization problem. Instead, from Theorem 4.1, an iterative procedure can be utilized to solve (4.2), along the lines of the pseudo-code summarized in Fig. 4.1. In the preprocessing step, the

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$, and the desired tensor subspace dimensionality $\{P_n, n = 1, 2, ..., N\}$.

Output: The N projection matrices $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, 2, ..., N\}$ that maximize the total scatter in the projected space.

Algorithm:

Step 1 (Preprocessing): Center the input samples as $\{\tilde{\mathcal{X}}_m = \mathcal{X}_m - \bar{\mathcal{X}}, m = 1, ..., M\}$, where $\bar{\mathcal{X}} = \frac{1}{M}\sum_{m=1}^{M} \mathcal{X}_m$ is the sample mean.

Step 2 (Initialization): Calculate the eigen-decomposition of $\mathbf{\Phi}^{(n)*} = \sum_{m=1}^{M} \tilde{\mathbf{X}}_{m(n)} \cdot \tilde{\mathbf{X}}_{m(n)}^T$ and set $\tilde{\mathbf{U}}^{(n)}$ to be consisting of the eigenvectors corresponding to the most significant P_n eigenvalues, for $n = 1, ..., N$.

Step 3 (Local optimization):

- Calculate $\{\tilde{\mathcal{Y}}_m = \tilde{\mathcal{X}}_m \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}, m = 1, ..., M\}$.
- Calculate $\Psi_{\mathcal{Y}_0} = \sum_{m=1}^{M} \parallel \tilde{\mathcal{Y}}_m \parallel_F^2$ (the mean $\bar{\tilde{\mathcal{Y}}}$ is all zero since $\tilde{\mathcal{X}}_m$ is centered).
- For $k = 1 : K$
  - For $n = 1 : N$
    - Set the matrix $\tilde{\mathbf{U}}^{(n)}$ to be consisting of the P_n eigenvectors of the matrix $\mathbf{\Phi}^{(n)}$, as defined in (4.3), corresponding to the largest P_n eigenvalues.
  - Calculate $\{\tilde{\mathcal{Y}}_m, m = 1, ..., M\}$ and $\Psi_{\mathcal{Y}_k}$.
  - If $(\Psi_{\mathcal{Y}_k} - \Psi_{\mathcal{Y}_{k-1}})/\Psi_{\mathcal{Y}_{k-1}} < \eta$, break and output $\{\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}, n = 1, 2, ..., N\}$.

Figure 4.1: The pseudo-code implementation of the proposed MPCA algorithm.

input tensors are centered first: $\{\tilde{\mathcal{X}}_m = \mathcal{X}_m - \bar{\mathcal{X}}, m = 1, ..., M\}$. The projection matrices are then initialized through the full projection truncation to be discussed in detail in Sec. 4.3.2. Next, in the local optimization step, the projection matrices are updated one by one (the "n loop") with all the others fixed. The local optimization procedure is

repeated (the "k loop") until the result converges or a maximum number K of iterations is reached. Each iteration consists of the N conditional subproblems.

4.2.3 Connections with existing solutions

From Sections 2.3.1 (page 27), 3.4.1 (page 50), and 2.2.4 (page 25), it can be seen that the MPCA introduced here generalizes PCA, 2DPCA, and GPCA.

When $N = 1$, PCA is simply MPCA with $N = 1$, where the input samples are vectors $\{\mathbf{x}_m \in \mathbb{R}^{I_1}\}$ with only one mode. Consequently, only one projection matrix $\mathbf{U}$ is needed in order to obtain the projected sample

$$\mathbf{y}_m = \mathbf{x}_m \times_1 \mathbf{U} = \mathbf{U}^T \mathbf{x}_m. \tag{4.5}$$

In this case, there is only one scatter matrix

$$\mathbf{\Phi}^{(n)} = \mathbf{\Phi}^{(1)} = \sum_{m=1}^{M} (\mathbf{x}_m - \bar{\mathbf{x}}) \cdot (\mathbf{x}_m - \bar{\mathbf{x}})^T, \tag{4.6}$$

which is the total scatter matrix in PCA. The optimal $\mathbf{U}$ is determined from the eigenvectors of $\mathbf{\Phi}^{(1)}$. Thus, MPCA subsumes PCA.

When $N = 2$, the input samples are matrices $\{\mathbf{X}_m \in \mathbb{R}^{I_1 \times I_2}\}$. From the review in Sec. 3.4.1 (page 50), the 2DPCA algorithm [153] is MPCA with $N = 2$ and a fixed $\mathbf{U}^{(1)} = \mathbf{I}$, where $\mathbf{I}$ is an identity matrix of size $I_1 \times I_1$. In this case, the projection becomes

$$\mathbf{Y}_m = \mathbf{X}_m \times_1 \mathbf{I} \times_2 \mathbf{U}^T = \mathbf{I}^T \mathbf{X}_m \mathbf{U} = \mathbf{X}_m \mathbf{U} \in \mathbb{R}^{I_1 \times P_2} \tag{4.7}$$

and only the 2-mode projection matrix needs to be solved. Similarly from the review in Sec. 3.4.1 (page 50), the GPCA algorithm [156] is MPCA with $N = 2$ and two projections matrices need to be solved.

4.3 Design and Computational Issues in MPCA

In this section, several issues pertinent to the implementation of MPCA are discussed. First, in-depth understanding of MPCA is provided. The properties of full projection are analyzed, and the geometric interpretation of the n-mode eigenvalues is introduced together with the concept of eigentensor. Next, the initialization method, the projection order determination, and the construction of termination criteria are described. Convergence issues are also discussed. Lastly, methods for subspace dimensionality determination are proposed, followed by the computational issues.

4.3.1 Full projection

With respect to this analysis, the term full projection refers to the multilinear projection for MPCA with $P_n = I_n$ for $n = 1, ..., N$. In this case, $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}^T_{\mathbf{\Phi}^{(n)}}$ is an identity matrix, as it can be seen from the following lemma:

Lemma 4.1. *When $P_n = I_n$ for $n = 1, ..., N$, $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}^T_{\mathbf{\Phi}^{(n)}}$ is an identity matrix.*

Proof. The proof is given in Appendix A.2. □

As a result, $\mathbf{\Phi}^{(n)}$ reduces to

$$\mathbf{\Phi}^{(n)*} = \sum_{m=1}^{M} \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right) \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right)^T. \tag{4.8}$$

In this case, $\mathbf{\Phi}^{(n)*}$ is determined by the input tensor samples only and it is independent of other projection matrices. The optimal $\tilde{\mathbf{U}}^{(n)} = \mathbf{U}^{(n)*}$ is then obtained as the matrix comprised of the eigenvectors of $\mathbf{\Phi}^{(n)*}$ directly without iteration, and the total scatter $\Psi_{\mathcal{X}}$ in the original data is fully captured. However, there is no dimensionality reduction through this full projection. From the properties of eigen-decomposition, it can be concluded that if all eigenvalues (per mode) are distinct, the full projection matrices

(corresponding eigenvectors) are also distinct and that the full projection is unique (up to sign) [40].

To interpret the geometric meanings of the n-mode eigenvalues, the total scatter tensor $\mathcal{Y}^*_{var} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ of the full projection is introduced as an extension of the total scatter matrix [3]. Each entry of the tensor $\mathcal{Y}^*_{var}$ is defined as below:

$$\mathcal{Y}^*_{var}(i_1, i_2, ..., i_N) = \sum_{m=1}^{M} \left[(\mathcal{Y}^*_m - \bar{\mathcal{Y}}^*)(i_1, i_2, ..., i_N) \right]^2, \tag{4.9}$$

where $\mathcal{Y}^*_m = \mathcal{X}_m \times_1 \mathbf{U}^{(1)*^T} \ldots \times_N \mathbf{U}^{(N)*^T}$ and $\bar{\mathcal{Y}}^* = \frac{1}{M} \sum_{m=1}^{M} \mathcal{Y}^*_m$. Using the above definition, it can be shown that for the so-called full projection ($P_n = I_n$ for all n), the i_nth n-mode eigenvalue $\lambda^{(n)*}_{i_n}$ is the sum of all the entries of the i_nth n-mode slice of $\mathcal{Y}^*_{var}$.

$$\lambda^{(n)*}_{i_n} = \sum_{i_1=1}^{I_1} \ldots \sum_{i_{n-1}=1}^{I_{n-1}} \sum_{i_{n+1}=1}^{I_{n+1}} \ldots \sum_{i_N=1}^{I_N} \mathcal{Y}^*_{var}(i_1, ..., i_{n-1}, i_n, i_{n+1}, ..., i_N). \tag{4.10}$$

In this dissertation, the eigenvalues are all arranged in a descending order. Figure 4.2 shows visually what the n-mode eigenvalues represent. In this figure, a number of third-order tensors, e.g. short sequences (3 frames) of images with size 5×4, are projected to a tensor space of size $5 \times 4 \times 3$ (full projection) so that a total scatter tensor $\mathcal{Y}^*_{var} \in \mathbb{R}^{5 \times 4 \times 3}$ is obtained.

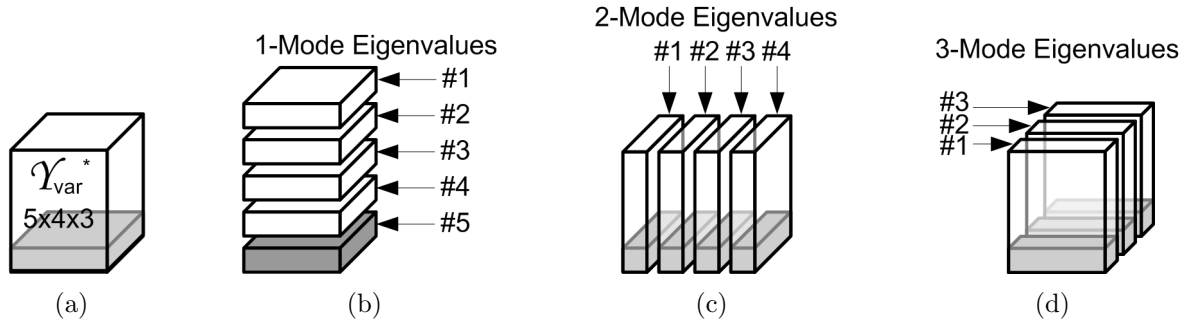


Figure 4.2: Visual illustration of: (a) the total scatter tensor, (b) the 1-mode eigenvalues, (c) 2-mode eigenvalues, and (d) the 3-mode eigenvalues in MPCA.

Using Equation (2.10) (page 19), each tensor $\mathcal{X}_m$ can be written as a linear combination of $P_1 \times P_2 \times ... \times P_N$ rank-1 tensors

$$\tilde{\mathcal{U}}_{p_1 p_2 ... p_N} = \tilde{\mathbf{u}}^{(1)}_{p_1} \circ \tilde{\mathbf{u}}^{(2)}_{p_2} \circ ... \circ \tilde{\mathbf{u}}^{(N)}_{p_N}. \tag{4.11}$$

These rank-1 tensors will be called, hereafter, eigentensors. Thus, the projected tensor $\mathcal{Y}^*_m$ can be viewed as the projection onto these eigentensors, with each entry of $\mathcal{Y}^*_m$ corresponding to one eigentensor. These definitions and illustrations help the understanding of MPCA in the following discussions.

4.3.2 Initialization by full projection truncation

Full projection truncation is used to initialize the iterative solution for MPCA, where the first P_n columns of the full projection matrix $\mathbf{U}^{(n)*}$ is kept to give an initial projection matrix $\tilde{\mathbf{U}}^{(n)}$. The corresponding total scatter is denoted as $\Psi_{\mathcal{Y}_0}$. This initialization is equivalent to the HOSVD-based solution in [79]. Although this full projection truncation initialization is not the optimal solution to (4.2), it is bounded and is considered a good starting point for the iterative procedure, as will be discussed below.

Remark 4.1. *There are other choices of initialization such as the truncated identity matrices [156, 150, 37] (named as pseudo identity matrices) and random matrices. Simulation studies (reported in Section 4.6) indicate that although in practical applications, the initialization step may not have a significant impact in terms of performance, it can affect the speed of convergence of the iterative solution. Since full projection truncation results in much faster convergence, it is used for MPCA initialization.*

In studying the optimality of the initialization procedure with respect to (4.2), assume, without loss of generality, that the 1-mode eigenvectors are truncated, in other words, only the first $P_1 < I_1$ 1-mode eigenvectors are kept. In this case, the following theorem applies.

Theorem 4.2. *Let* $\mathbf{U}^{(n)*}$ *and* $\lambda_{i_n}^{(n)*}$, $i_n = 1, ..., I_n$, *be the matrix of the eigenvectors of* $\mathbf{\Phi}^{(n)*}$ *and the eigenvalues of* $\mathbf{\Phi}^{(n)*}$, *respectively, and* $\tilde{\mathcal{Y}}_m = \tilde{\mathcal{X}}_m \times_1 \mathbf{U}^{(1)*^T} ... \times_N \mathbf{U}^{(N)*^T}, m = 1, ..., M$. *Keep only the first* $P_1 < I_1$ *eigenvectors with* $\sum_{i_1=P_1+1}^{I_1} \lambda_{i_1}^{(1)*} > 0$ *to get* $\check{\mathcal{X}}_m = \check{\mathcal{Y}}_m \times_1 \check{\mathbf{U}}^{(1)} \times_2 \mathbf{U}^{(2)*} ... \times_N \mathbf{U}^{(N)*}$, *where* $\check{\mathcal{Y}}_m = \tilde{\mathcal{Y}}_m(1 : P_1, :, ..., :)$ *and* $\check{\mathbf{U}}^{(1)} = \mathbf{U}^{(1)*}(:, 1 : P_1)$. *Let* $\check{\mathbf{\Phi}}^{(n)}$ *correspond to* $\check{\mathcal{X}}_m$, *and the matrix of its eigenvectors and its eigenvalues be* $\hat{\mathbf{U}}^{(n)}$ *and* $\hat{\lambda}_{i_n}^{(n)}$, *respectively. Then,*

$$\hat{\lambda}_{i_1}^{(1)} = \begin{cases} \lambda_{i_1}^{(1)*}, & i_1 = 1, ..., P_1 \\ 0, & i_1 = P_1 + 1, ..., I_1. \end{cases}$$

For $n > 1$ *(other modes),* $\hat{\lambda}_{i_n}^{(n)} \leq \lambda_{i_n}^{(n)*}$. *Furthermore, for each mode, at least for one value of* i_n, $\hat{\lambda}_{i_n}^{(n)} < \lambda_{i_n}^{(n)*}$.

Proof. The proof is given in Appendix A.3. □

It can be seen from Theorem 4.2 that if a non-zero eigenvalue is truncated in one mode, the eigenvalues in all the other modes tend to decrease in magnitude and the corresponding eigenvectors change accordingly. Thus, the eigen-decomposition needs to be recomputed in all the other modes, i.e., the projection matrices in all the other modes need to be updated. Since from Theorem 4.1, the computations of all the projection matrices are inter-dependent, the update of a projection matrix $\tilde{\mathbf{U}}^{(n^*)}$ updates the matrices $\{\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}, n \neq n^*\}$ as well. Consequently, the projection matrices in all the other modes $\{\tilde{\mathbf{U}}^{(n)}, n \neq n^*\}$ are no longer consisting of the eigenvectors of the corresponding (updated) $\mathbf{\Phi}^{(n)}$ and they need to be updated. The update continues until the termination criterion, discussed in Sec. 4.3.4, is satisfied.

Figure 4.2 provides a visual illustration of Theorem 4.2. Removal of a basis vector in one mode results in eliminating a slice of $\mathcal{Y}_{var}^*$. In Fig. 4.2, if the last non-zero (fifth) 1-mode eigenvalue is discarded (shaded in Fig. 4.2(b)), the corresponding (fifth) 1-mode slice of $\mathcal{Y}_{var}^*$ is removed (shaded in Fig. 4.2(a)), resulting in a truncated total scatter

tensor $\tilde{\mathcal{Y}}^*_{var} \in \mathbb{R}^{4\times4\times3}$. Discarding this slice will affect all eigenvalues in the remaining modes, whose corresponding slices have a non-empty overlap with the discarded 1-mode slice. In Figs. 4.2(c) and 4.2(d), the shaded part indicates the removed 1-mode slice corresponding to the discarded eigenvalue.

Having proven the non-optimality of full projection truncation with respect to the objective function (4.2), the bounds for full projection truncation are then derived in the following theorem.

Theorem 4.3. *Let $\lambda^{(n)*}_{i_n}$ denote the i_nth n-mode eigenvalue for the n-mode full projection matrix. The upper and lower bounds for $(\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0})$, the loss of variation due to the full projection truncation (measured by the total scatter), are derived as follows:*

$$\Psi_L = \max_n \sum_{i_n=P_n+1}^{I_n} \lambda^{(n)*}_{i_n} \leq (\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0}) \leq \sum_{n=1}^{N} \sum_{i_n=P_n+1}^{I_n} \lambda^{(n)*}_{i_n} = \Psi_U. \tag{4.12}$$

Proof. The proof is given in Appendix A.4. □

From (4.12), it can be seen that the tightness of the bounds is determined by the eigenvalues in each mode. The bounds can be observed in Fig. 4.2. For instance, truncation of the last eigenvector in each of the three modes results in another truncated total scatter tensor $\hat{\mathcal{Y}}^*_{var} \in \mathbb{R}^{4\times3\times2}$. Thus, the difference between $\Psi_{\mathcal{X}}$ and $\Psi_{\mathcal{Y}_0}$ (the sum of all entries in $\mathcal{Y}^*_{var}$ and $\hat{\mathcal{Y}}^*_{var}$, respectively) is upper-bounded by the total of the sums of all the entries in each truncated slice and lower-bounded by the maximum sum of all the entries in each truncated slice. For full projection truncation, the gap between the actual loss of variation and the upper bound is due to the multiple counts of the overlaps between the discarded slice in one mode and the discarded slices in the other modes of $\mathcal{Y}^*_{var}$.

The tightness of the bounds Ψ_U and Ψ_L depends on the order N, the eigenvalue characteristics (distribution) such as the number of zero-valued eigenvalues, and the degree of truncation P_n. For example, for $N = 1$, the case of PCA, $\Psi_L = \Psi_U$ and the

full projection truncation is the optimal solution so no iterations are necessary. A larger N results in more terms in the upper bound and tends to lead to looser bound, and vice versa. In addition, if all the truncated eigenvectors correspond to zero-valued eigenvalues, $\Psi_{\mathcal{Y}_0} = \Psi_{\mathcal{X}}$ since $\Psi_L = \Psi_U = 0$, and the full projection truncation results in the optimal solution.

4.3.3 Projection order

The MPCA algorithm computes the N projection matrices in a certain order and this order may affect the obtained solution. Thus, the effects of the projection order have been studied empirically in this work and simulation results presented in Section 4.6 indicate that altering the ordering of the projection matrix computation does not result in significant performance differences in practical situations. Therefore, a sequential order from 1 to N is taken in implementation.

4.3.4 Termination

The termination criterion can be determined using the objective function $\Psi_{\mathcal{Y}}$. In particular, the iterative procedure terminates if $(\Psi_{\mathcal{Y}_k} - \Psi_{\mathcal{Y}_{k-1}})/\Psi_{\mathcal{Y}_{k-1}} < \eta$, where $\Psi_{\mathcal{Y}_k}$ and $\Psi_{\mathcal{Y}_{k-1}}$ are the resulted total scatter from the kth and $(k-1)$th iterations, respectively, and η is a user-defined small number threshold (e.g., $\eta = 10^{-6}$). In other words, the iterations stop if there is little improvement in the resulted total scatter. In practice, for computational consideration, another easy way to terminate the iteration is to set the maximum number of iterations allowed to K.

4.3.5 Convergence of the MPCA algorithm

The derivation of Theorem 4.1 (Appendix A.1) implies that per iteration, the total scatter $\Psi_{\mathcal{Y}}$ is a non-decreasing function (it either remains the same or increases) since each update

of the projection matrix $\tilde{\mathbf{U}}^{(n^*)}$ in a given mode n^* maximizes $\Psi_{\mathcal{Y}}$. On the other hand, $\Psi_{\mathcal{Y}}$ is upper-bounded by $\Psi_{\mathcal{X}}$ (the variation in the original samples) since the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$ consist of orthonormal columns. Therefore, MPCA is expected to have good convergence property. Empirical results presented in Section 4.6 indicate that the proposed MPCA algorithm converges very fast (within 5 iterations) for typical tensor objects. Furthermore, when per mode eigenvalues are all distinct (with multiplicity 1), which is the case for the simulated data as well as the face and gait data, the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$, which maximize $\Psi_{\mathcal{Y}}$, are expected to converge as well. It should be noted that the claimed convergence regarding the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$ is under the condition that the sign for the first component of each n-mode eigenvector is fixed since the eigenvector is unique up to sign. Simulation studies show that the projection matrices $\{\tilde{\mathbf{U}}^{(n)}\}$ do converge within a small number of iterations.

4.3.6 Determination of subspace dimensionality

When the targeted dimensionality $\{P_n, n = 1, ..., N\}$ is not specified in advance, its value has to be determined before solving the MPCA projection. Consequently, the objective function (4.2) needs to be revised to include a constraint on the desired dimensionality reduction. The revised objective function is as follows:

$$\{\tilde{\mathbf{U}}^{(n)}, P_n, n = 1, ..., N\} = \arg \max_{\tilde{\mathbf{U}}^{(1)},...,\tilde{\mathbf{U}}^{(N)},P_1,...,P_N} \Psi_{\mathcal{Y}}, \quad \texttt{subject to } \frac{\prod_{n=1}^{N} P_n}{\prod_{n=1}^{N} I_n} < \Delta, \tag{4.13}$$

where the ratio between the targeted (reduced) dimensionality and the original tensor space dimensionality is utilized to measure the amount of dimensionality reduction, and Δ is a threshold parameter to be specified by the user or determined based on empirical studies.

The first proposed subspace dimensionality determination solution is called sequential mode truncation. Starting with $P_n = I_n$ for all n at $\tau = 0$, at each subsequent step

$\tau = \tau + 1$, the sequential mode truncation truncates, in a selected mode n, the P_nth n-mode eigenvector of the reconstructed input tensors. The truncation could be interpreted as the elimination of the corresponding P_nth n-mode slice of the total scatter tensor. For the mode selection, the scatter loss rate $\vartheta_\tau^{(n)}$ due to the truncation of its P_nth eigenvector is calculated for each mode. $\vartheta_\tau^{(n)}$ is defined as follows:

$$\vartheta_\tau^{(n)} = \frac{\Psi_{\mathcal{Y}_{(\tau)}} - \Psi_{\mathcal{Y}_{(\tau-1)}}}{\left[P_n \cdot \prod_{j=1,j\neq n}^{N} P_j\right] - \left[(P_n - 1) \cdot \prod_{j=1,j\neq n}^{N} P_j\right]} = \frac{\tilde{\lambda}_{P_n}^{(n)}}{\Pi_{j=1,j\neq n}^{N} P_j}, \tag{4.14}$$

where $\Psi_{\mathcal{Y}_{(\tau)}}$ is the scatter obtained at step τ, $\Pi_{j=1,j\neq n}^{N} P_j$ is the amount of dimensionality reduction achieved, and $\tilde{\lambda}_{P_n}^{(n)}$, the corresponding P_nth n-mode eigenvalue, is the loss of variation due to truncating the P_nth n-mode eigenvector. The mode with the smallest $\vartheta_\tau^{(n)}$ is selected for the step-τ truncation. For the selected mode n, P_n is decreased by 1: $P_n = P_n - 1$ and $\frac{\prod_{n=1}^{N} P_n}{\prod_{n=1}^{N} I_n} < \Delta$ is tested. The truncation stops when $\frac{\prod_{n=1}^{N} P_n}{\prod_{n=1}^{N} I_n} < \Delta$ is satisfied. Otherwise, the input tensors are reconstructed according to (2.7) (page 18) using the current truncated projection matrices and they are used to recompute the n-mode eigenvalues and eigenvectors corresponding to full projection. Since eigenvalues in other modes are affected by the eigenvector truncation in a given mode (see Theorem 4.2), it is expected that the sequential mode truncation, which takes into account this effect, constitutes a reasonable good choice for determining P_n in the sense of (4.13).

Beside the method of sequential mode truncation, the Q-based method, a suboptimal, simplified dimensionality determination procedure that requires no recomputation, is also proposed for use in practice. Define the ratio

$$Q^{(n)} = \frac{\sum_{i_n=1}^{P_n} \lambda_{i_n}^{(n)*}}{\sum_{i_n=1}^{I_n} \lambda_{i_n}^{(n)*}} \tag{4.15}$$

to be the remained portion of the total scatter in the n-mode after the truncation of the n-mode eigenvectors beyond the P_nth, where $\lambda_{i_n}^{(n)*}$ is the i_nth full-projection n-mode

eigenvalue. In the proposed Q-based method, the first P_n eigenvectors are kept in the n-mode (for each n) so that: $Q^{(1)} = Q^{(2)} = ... = Q^{(N)} = Q$ (the equality can hold approximately since it is unlikely to find P_n that gives the exact equality in practice). It should be noted that $\sum_{i_n=1}^{I_n} \lambda_{i_n}^{(n)*} = \Psi_{\mathcal{X}}$ for all n since from Theorem 4.1, the total scatter for the full projection was given as:

$$\Psi_{\mathcal{Y}}^{*} = \Psi_{\mathcal{X}} = \sum_{m=1}^{M} \| \mathbf{Y}_{m(n)} - \bar{\mathbf{Y}}_{(n)} \|_F^2 = \sum_{i_n=1}^{I_n} \lambda_{i_n}^{(n)*}, n = 1, ..., N. \tag{4.16}$$

This method can be viewed as an extension of the dimensionality selection strategy in the traditional PCA to the multilinear case. The reason behind this choice is that loss of variation is (approximately) proportional to the sum of the corresponding eigenvalues of the discarded eigenvectors. By discarding the least significant eigenvectors in each mode, the variation loss can be contained and a tighter lower bound for $\Psi_{\mathcal{Y}_0}$ is obtained. The empirical study to be reported in the experimental section indicates that the Q-based method provides results similar to those obtained by sequential mode truncation (as measured in terms of the total scatter captured). Thus, it can be safely used instead of the more computationally expensive sequential mode truncation.

4.3.7 Computational issues

Apart from the actual performance of MPCA, its computational complexity, memory requirements, and storage needs are relative measures of its practicality and usefulness as they determine the required computing power and processing (execution) time. Here, MPCA-related computational issues are examined in a fashion similar to that introduced in [156].

Since the MPCA solution is iterative, the computational complexity analysis is performed for one iteration. For simplicity, it is assumed that $I_1 = I_2 = ... = I_N = \left(\prod_{n=1}^{N} I_n\right)^{\frac{1}{n}} = I$. From a computational complexity point of view, the most demanding

steps are the formation of the matrices $\mathbf{\Phi}^{(n)}$, the eigen-decomposition of $\mathbf{\Phi}^{(n)}$, and the computation of the multilinear projection $\tilde{\mathcal{Y}}_m$. It should be noted that the use of multilinear multiplication and unfolding in order to compute $\mathbf{\Phi}^{(n)}$ is more efficient comparing to the use of Kronecker products. The computations needed to determine $\mathbf{\Phi}^{(n)}$, the P_n eigenvectors of $\mathbf{\Phi}^{(n)}$, and $\tilde{\mathcal{Y}}_m$ are in order of $O(MN \cdot I^{(N+1)})$ (upper bounded), $O(I^3)$, and $O(N \cdot I^{(N+1)})$, respectively. The total complexity is

$$O((N+1) \cdot M \cdot N \cdot I^{(N+1)} + N \cdot I^3). \tag{4.17}$$

If the algorithm is terminated by setting K rather than checking the convergence, the computation of $\tilde{\mathcal{Y}}_m$ is not needed and the total complexity becomes

$$O(N^2 \cdot M \cdot I^{(N+1)} + N \cdot I^3). \tag{4.18}$$

In MPCA, $\bar{\mathcal{X}}$ and $\mathbf{\Phi}^{(n)}$ can be computed incrementally by reading $\mathcal{X}_m$ or $\tilde{\mathcal{X}}_m$ sequentially without loss of information. Hence, memory requirements for the MPCA algorithm can be as low as $O(\prod_{n=1}^{N} I_n)$ as MPCA computes the solution without requiring all data samples in the memory. This is a major advantage that MPCA enjoys over the HOSVD-based solutions [64, 79], which requires the formation of an $(N+1)$th-order tensor when the input tensor samples are of Nth-order. This is of considerable importance in applications with large data sets as the size of the input database may lead to significant increase in complexity and memory storage requirement. On the other hand, as an iterative solution, MPCA has a higher I/O cost than a non-iterative solution. Nevertheless, since solving for the projection matrices using MPCA is only in the training phase of the targeted recognition tasks, it can be done offline and the additional I/O (and computational) cost due to iterations are not considered a disadvantage of the proposed MPCA solution.

MPCA compresses each tensor sample of size $\prod_{n=1}^{N} I_n$ to $\prod_{n=1}^{N} P_n$, and it needs N

matrices $\tilde{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times P_n}$ for compression and decompression. Thus, it requires

$$\left(M \cdot \prod_{n=1}^{N} P_n + \sum_{n=1}^{N} (I_n \times P_n) \right) \tag{4.19}$$

scalars in the reduced space by MPCA and the compression ratio is defined as:

$$CR = \frac{M \cdot \prod_{n=1}^{N} I_n}{M \cdot \prod_{n=1}^{N} P_n + \sum_{n=1}^{N} (I_n \times P_n)}. \tag{4.20}$$

In studying the subspace dimensionality determination performance in the experiments (Sec. 4.6), algorithms are compared under the same CR.

4.4 Discriminative MPCA Feature Selection and MPCA+LDA

The projection matrices $\{\tilde{\mathbf{U}}^{(n)}, n = 1, ..., N\}$ obtained by MPCA can be used to extract features from a set of training tensor samples $\{\mathcal{X}_m, m = 1, ..., M\}$. In testing, a normalized tensor sample $\mathcal{X}$ is centered by subtracting the mean obtained from the gallery tensors and then projected to the MPCA feature $\mathcal{Y}$:

$$\mathcal{Y} = (\mathcal{X} - \bar{\mathcal{X}}) \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T}. \tag{4.21}$$

From the gallery set, a set of eigentensors is obtained with reduced dimensionality ($P_n \leq I_n$) determined by a user-specified Q, and each entry of a projected tensor feature can be viewed as a (scalar) feature corresponding to a particular eigentensor. Some of the small variation and noise are removed in the projection. However, for recognition, it should be noted that MPCA is an unsupervised technique without taking class labels into account. As a result, the variation captured in the projected tensor subspace includes both the within-class variation and the between-class variation. In the task of classi-

fication, a larger between-class variation relative to the within-class variation indicates good class separability, while a smaller between-class variation relative to the within-class variation indicates poor class separability. Hence, a feature selection strategy is proposed to select eigentensors according to their class discrimination power [152, 3, 139], defined to be the ratio of the between-class scatter over the within-class scatter:

Definition 4.1. *The class discriminability* $\Theta_{p_1p_2\ldots p_N}$ *for an eigentensor* $\tilde{\mathcal{U}}_{p_1p_2\ldots p_N}$ *is defined as*

$$\Theta_{p_1p_2\ldots p_N} = \frac{\sum_{c=1}^{C} M_c \cdot \left[\bar{\mathcal{Y}}_c(p_1,p_2,...,p_N) - \bar{\mathcal{Y}}(p_1,p_2,...,p_N)\right]^2}{\sum_{m=1}^{M} \left[\mathcal{Y}_m(p_1,p_2,...,p_N) - \bar{\mathcal{Y}}_{c_m}(p_1,p_2,...,p_N)\right]^2}, \tag{4.22}$$

where C *is the number of classes,* M *is the number of samples in the gallery set,* M_c *is the number of samples for class* c *and* c_m *is the class label for the* m*th gallery sample* $\mathcal{X}_m$*.* $\mathcal{Y}_m$ *is the feature tensor of* $\mathcal{X}_m$ *in the projected tensor subspace. The mean feature tensor* $\bar{\mathcal{Y}} = \frac{1}{M}\sum_m \mathcal{Y}_m$ *and the class mean feature tensor* $\bar{\mathcal{Y}}_c = \frac{1}{M_c}\sum_{m,c_m=c} \mathcal{Y}_m$*.*

For the eigentensor selection, the entries in the projected tensor $\mathcal{Y}_m$ are rearranged into a feature vector $\mathbf{y}_m$, ordered according to $\Theta_{p_1p_2\ldots p_N}$ in descending order, and only the first $H_{\mathbf{y}}$ most discriminative components of $\mathbf{y}_m$ are kept for classification, with $H_{\mathbf{y}}$ determined empirically or user-specified. By this selection, a more discriminating subspace is resulted compared to the MPCA projected tensor subspace that includes both features with good separability and features with poor separability. Next, a weight tensor $\mathcal{W}$ is formed with entries defined as $\mathcal{W}(p_1,p_2,...,p_N) = \sqrt{\Pi_{n=1}^{N}\lambda_{p_n}^{(n)}}$, where $\lambda_{p_n}^{(n)}$ denotes the p_nth n-mode eigenvalue corresponding to the projection matrix $\tilde{\mathbf{U}}^{(n)}$. $\mathcal{W}$ is rearranged into a vector $\mathbf{w}$ in the same order as $\mathbf{y}_m$, with only the first $H_{\mathbf{y}}$ components kept and $\mathbf{w}$ can be used as weights in measuring distances (Table 3.1, page 40).

The feature vector $\mathbf{y}_m$ can be used directly for recognition, or LDA can also be applied to obtain an MPCA+LDA approach for recognition, similar to the popular approach of PCA+LDA [3]. Let $\mathbf{S}_{B_{\mathbf{y}}}$ and $\mathbf{S}_{W_{\mathbf{y}}}$ be the between-class scatter matrix and within-class scatter matrix based on $\{\mathbf{y}_m\}$, respectively. Then, from Section 2.3.2 (page 28),

the corresponding LDA projection $\mathbf{U}_{LDA}$ consists of the first $H_{\mathbf{z}}$ ($\leq C-1$) generalized eigenvectors of the following generalized eigenvalue problem: $\mathbf{S}_{B_{\mathbf{y}}}\mathbf{u}_{h_{\mathbf{z}}} = \lambda_{h_{\mathbf{z}}}\mathbf{S}_{W_{\mathbf{y}}}\mathbf{u}_{h_{\mathbf{z}}}$, where $h_{\mathbf{z}} = 1, ..., H_{\mathbf{z}}$ and $\lambda_{h_{\mathbf{z}}}$ is the $h_{\mathbf{z}}$th largest generalized eigenvalue. Thus, the MPCA+LDA feature vector $\mathbf{z}_m$ is obtained as: $\mathbf{z}_m = \mathbf{U}_{LDA}^T\mathbf{y}_m$.

As discussed in Section 1.3 (page 4), in practical face or gait recognition problems, many factors, such as pose, illumination, expression, viewing angles, walking surfaces, and shoes, may affect a person's face or gait. Thus, the face or gait patterns in practice are expected to be highly nonlinear and complex. Moreover, the face or gait data available for training and testing may be captured under different conditions and good generalization is very difficult. The MPCA and MPCA+LDA algorithms proposed so far have not taken into consideration of these complex and nonlinear pattern distributions, and in the next section, the combination of MPCA with ensemble-based learning is proposed to further improve the generalization performance on recognition problems.

4.5 Boosting LDA on the MPCA Features (B-LDA-MPCA)

There are many methods proposed in the literature to handle complex and nonlinear patterns. The ensemble-based machine learning method called boosting is a very promising one offering good generalization capability through combining a set of weak learners repeatedly trained on weighted training samples [26, 111]. A short review of the popular Adaptive Boosting (AdaBoost) algorithm is provided in Appendix B. Boosting requires an appropriate weak learner to work, which has restricted its applicability [111, 120]. A recent work in [93] has broken this limitation by proposing a boosting algorithm that works with LDA-style learners. A cross-validation mechanism is employed to weaken the LDA learner and the pairwise class discriminant distribution is introduced for interaction between the booster and the learner.

This section investigates the combination of MPCA with the LDA-based boosting work in [93]. In this novel processing scheme, MPCA is first used to generate eigentensors in a lower-dimensional tensor space and then only a number of discriminative eigentensors are selected as the input to the LDA-based booster. This eigentensor number provides one more way (besides the cross-validation mechanism in [93]) to control the weakness of the LDA learner, and the MPCA feature extractor before the booster greatly reduces the processing cost (in both training and testing) so that very-high dimensional tensorial data can be handled efficiently. Furthermore, a novel regularization control mechanism is added to the LDA learners to reduce overfitting on the gallery set and improve the generalization as the within-class scatter of testing patterns is often expected to be larger than that of the training patterns.

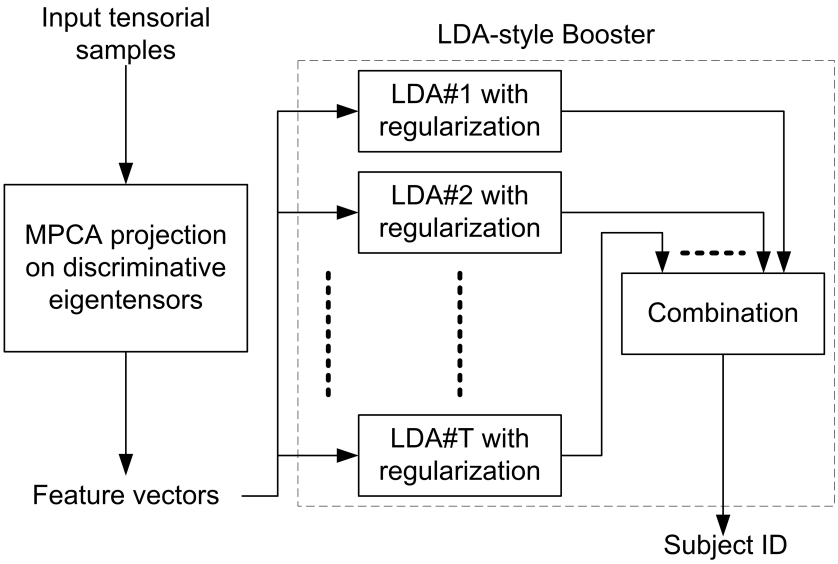


Figure 4.3: Illustration of recognition through boosting LDA with regularization on MPCA features.

The block diagram of the proposed combination of MPCA and boosting is shown in Figure 4.3. Input tensorial samples are projected on a number of discriminative eigentensors to obtain feature vectors, as described in Section 4.4, and these vectors are fed into the LDA-based booster for learning and classification. It should be noted that the booster proposed here has an important difference with that in [93]. The LDA-style

base learners in the proposed booster take $\{\mathbf{y}_m \in \mathbb{R}^{H_\mathbf{y}}, m = 1, ..., M\}$, the feature vectors extracted by MPCA, rather than the vectorized original data $\{\mathbf{x}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}, m = 1, ..., M\}$ as in [93]. There are two benefits from the proposed scheme:

1. The feature vector dimension $H_\mathbf{y}$, the number of discriminative eigentensors selected, offers one more degree (besides ξ, the number of samples per class used for the LDA learners) to control the weakness of the LDA learners. Similar to the PCA+LDA, where the performance is often affected by the number of principal components selected for LDA, $H_\mathbf{y}$ affects the performance of LDA on the MPCA features as well. Therefore, by choosing a $H_\mathbf{y}$ that is not optimal for a single LDA learner, the obtained LDA learner is weakened. Of course, the LDA learner cannot be made "too weak" either. Otherwise, the boosting scheme will not work.

2. Using feature vectors of dimension $H_\mathbf{y}$ instead of the original data as the booster input is computationally advantageous. Since boosting is an iterative algorithm with T rounds, the computational cost is about T times of that of a single learner with the same input, both in training and testing. By making the booster to work on lower dimensional features extracted by MPCA, the booster becomes much more efficient since it only needs to deal with low-dimensional vectors. Consequently, the computational cost is reduced significantly.

The AdaBoost algorithm is developed for binary classification problems and several methods have been proposed to extend the AdaBoost to the multiclass case [27, 113, 1, 110]. The multi-class AdaBoost approach followed here is the AdaBoost.M2 algorithm [26]. The pseudo-code implementation of the proposed MPCA+boosting scheme is summarized in Fig. 4.4.

The AdaBoost.M2 aims to extend the communication between the boosting algorithm and the weak learner by allowing the weak learner to generate more expressive hypotheses (a set of "plausible" labels rather than a single label) indicating a "degree of

Input: The gallery feature vectors $\{\mathbf{y}_m, m = 1, ..., M\}$ with class labels $\mathbf{c} \in \mathbb{R}^M$, the LDA learner described in Sec. 4.5, the number of samples per class for LDA training ξ, the maximum number of iterations T.

Algorithm:

Initialize $\mathbf{D}_1(m, c) = \frac{1}{M(C-1)}$, $\hat{\mathbf{A}}_1(c_a, c_b) = \frac{1}{C^2}$, $\mathbf{D}_1(m, c_m) = 0$, $\hat{\mathbf{A}}_1(c_a, c_a) = 0$, and the first ξ samples from each class is selected to form the initial training set $\{\mathbf{y}_s, s = 1, ..., S\}_1$.

Do for $t = 1 : T$:

1. Get $\hat{\mathbf{U}}_{LDA_t}$ from $\mathbf{S}_{B_t}$ and $\mathbf{S}_{W_t}$ constructed from $\{\mathbf{y}_s, s = 1, ..., S\}_t$ and project $\{\mathbf{y}_m\}$ to $\{\hat{\mathbf{z}}_m\}$.
2. Get hypothesis $\{h_t(\mathbf{y}_m, c) \in [0, 1]\}$ by applying the nearest mean classifier on $\{\hat{\mathbf{z}}_m\}$.
3. Calculate $\hat{\epsilon}_t$, the pseudo-loss of h_t, from (4.23).
4. Set $\beta_t = \hat{\epsilon}_t/(1 - \hat{\epsilon}_t)$.
5. Update $\mathbf{D}_t$:

$$\mathbf{D}_{t+1}(m, c) = \mathbf{D}_t(m, c)\beta_t^{\frac{1}{2}(1+h_t(\mathbf{y}_m,c_m)-h_t(\mathbf{y}_m,c))},$$

   and normalize it:

$$\mathbf{D}_{t+1}(m, c) = \frac{\mathbf{D}_{t+1}(m, c)}{\sum_m \sum_c \mathbf{D}_{t+1}(m, c)}.$$

6. Update $\mathbf{d}_{t+1}(m)$, $\hat{\mathbf{A}}_{t+1}$, and $\{\mathbf{y}_s\}_{t+1}$ accordingly.

Output: The final hypothesis:

$$h_{fin}(\mathbf{y}) = \arg\max_c \sum_{t=1}^{T} \left(\log \frac{1}{\beta_t}\right) h_t(\mathbf{y}, c)$$

Figure 4.4: The pseudo-code implementation of the LDA-based booster on MPCA features.

plausibility", i.e., a hypothesis h takes a sample $\mathbf{y}$ and a class label c as the inputs and produces a "plausibility" score $h(\mathbf{y}, c) \in [0, 1]$ as the output. To achieve its objective, the AdaBoost.M2 introduces a sophisticated error measure pseudo-loss $\hat{\epsilon}_t$ with respect to the mislabel distribution $\mathbf{D}_t(m, c)$ in [26]. A mislabel is a pair (m, c), where m is the index of a training sample and c is an incorrect label associated with the sample $\mathbf{y}_m$. Let B be the set of all mislabels: $B = \{(m, c) : m = 1, ..., M, c \neq c_m\}$. The mislabel distribution is initialized as $\mathbf{D}_1(m, c) = \frac{1}{M \cdot (C-1)}$ for $(m, c) \in B$. Accordingly, the weak learner produces a hypothesis $h_t : \mathbb{R}^I \times C \rightarrow [0, 1]$, where $h(\mathbf{y}, c)$ measures the degree to which it is believed that c is the correct label for $\mathbf{y}$. The pseudo-loss $\hat{\epsilon}_t$ of the hypothesis h_t with respect to $\mathbf{D}_t(m, c)$ is defined to measure the goodness of h_t and it is given by [26]:

$$\hat{\epsilon_t} = \frac{1}{2} \sum_{(m,c) \in B} \mathbf{D}_t(m, c) \left(1 - h_t(\mathbf{y}_m, c_m) + h_t(\mathbf{y}_m, c)\right). \tag{4.23}$$

The introduction of the mislabel distribution enhances the communication between the learner and the booster, so that the AdaBoost.M2 can focus the weak learner not only on hard-to-classify samples, but also on the incorrect labels that are the hardest to discriminate [26].

Another distribution $\mathbf{d}_t(m)$, named as the pseudo sample distribution in [93], is derived from $\mathbf{D}_t(m, c)$ as $\mathbf{d}_t(m) = \sum_{c \neq c_m} \mathbf{D}_t(m, c)$. For the communication between the booster and the learner, the modified "pairwise class discriminant distribution" (PCDD) $\hat{\mathbf{A}}_t \in \mathbb{R}^{C \times C}$ introduced in [93] is employed as

$$\hat{\mathbf{A}}_t(c_a, c_b) = \frac{1}{2} \left(\sum_{c_m = c_a, c_{m_t} = c_b} \mathbf{d}_t(m) + \sum_{c_m = c_b, c_{m_t} = c_a} \mathbf{d}_t(m) \right), \tag{4.24}$$

where $c_{m_t} = \arg\max_c h_t(\mathbf{y}_m, c)$ and the diagonal of $\hat{\mathbf{A}}_t$ is set to zeros. This version of PCDD results in more independence and diversity between learners, which tends to achieve a low generalization error.

In building the LDA learner, the approach in [93] is adopted with several modifications. Firstly, only ξ samples per class are used as the input to the LDA learner in order to get weaker but more diverse LDA learners. The first ξ samples are taken for the first boosting step and the hardest ξ (with the largest $\mathbf{d}(m)$) samples are selected for subsequent steps. Let $\{\mathbf{y}_s, s = 1, ..., S\}_t$ denote the selected samples in round t, where $S = \xi \times C$. Next, for the between-class scatter matrix $\hat{\mathbf{S}}_B$, the pairwise between-class scatter in [74] is used instead of that used in [93] for its simplicity and easy computation:

$$\hat{\mathbf{S}}_B = \sum_{c_a=1}^{C-1} \sum_{c_b=c_a+1}^{C} \hat{\mathbf{A}}_t(c_a, c_b)(\bar{\mathbf{y}}_{c_a} - \bar{\mathbf{y}}_{c_b})(\bar{\mathbf{y}}_{c_a} - \bar{\mathbf{y}}_{c_b})^T, \tag{4.25}$$

where $\bar{\mathbf{y}}_c = \frac{1}{\xi} \sum_s^{c_s=c} \mathbf{y}_s$. Finally, for the within-class scatter matrix, a regularized version of that in [93] is used:

$$\hat{\mathbf{S}}_W = \sum_s \mathbf{d}(s)(\mathbf{y}_s - \bar{\mathbf{y}}_{c_s})(\mathbf{y}_s - \bar{\mathbf{y}}_{c_s})^T + \kappa \cdot \mathbf{I}_{H_\mathbf{y}}, \tag{4.26}$$

where κ is a regularization parameter to increase the estimated within-class scatter and $\mathbf{I}_{H_\mathbf{y}}$ is an identity matrix of size $H_\mathbf{y} \times H_\mathbf{y}$. The regularization term is added because in challenging face or gait recognition problems, the actual within-class scatter of testing (probe) samples is expected to be greater than the within-class scatter that can be estimated from the gallery set. With these definitions, the corresponding LDA projection $\hat{\mathbf{U}}_{LDA}$ consists of the first $H_{\hat{\mathbf{z}}}$ ($\leq C - 1$) generalized eigenvectors of the following generalized eigenvalue problem: $\hat{\mathbf{S}}_B \hat{\mathbf{u}}_{h_{\hat{\mathbf{z}}}} = \lambda_{h_{\hat{\mathbf{z}}}} \hat{\mathbf{S}}_W \hat{\mathbf{u}}_{h_{\hat{\mathbf{z}}}}$, where $h_{\hat{\mathbf{z}}} = 1, ..., H_{\hat{\mathbf{z}}}$ and $\lambda_{h_{\hat{\mathbf{z}}}}$ is the $h_{\hat{\mathbf{z}}}$th largest generalized eigenvalue. Thus, the LDA feature vector $\hat{\mathbf{z}}_m$ is obtained as $\hat{\mathbf{z}}_m = \hat{\mathbf{U}}_{LDA}^T \mathbf{y}_m$ for the input to a classifier. To produce the hypothesis, the nearest mean classifier (NMC), which assigns label c to the test sample $\mathbf{y}$ if $\bar{\mathbf{y}}_c$ is the class mean nearest to $\mathbf{y}$, is used and the calculated distances between a sample and the C class means are matched to the interval $[0, 1]$ as required by the AdaBoost.M2 algorithm.

4.6 Experimental Study

This section investigates the various properties of the MPCA algorithm. Detailed face and gait recognition results as well as comparisons with competing algorithms will be presented in Chapter 7 together with the other proposed algorithms. The MPCA properties studied are: a) the effects of the initial conditions, b) the effects of the ordering of the projection matrix computation, c) the convergence of the projection matrices, d) the evolution of the total scatter $\Psi_{\mathcal{Y}}$ over iterations, e) the number of iterations needed for convergence, and f) the performance of the tensor subspace dimensionality determination proposal. As many of these properties may be affected by the nature of the tensorial data, three synthetic data sets with different eigenvalue distributions are constructed for this study. Because the study results on the face or gait data are similar to one of the synthetic data sets, this section reports only the experiments performed on the synthetic data sets for illustration. In the following, the synthetic data generation method is described first. Then, the study on various MPCA properties is presented.

4.6.1 Synthetic data generation

The core of the MPCA algorithm is the eigen-decomposition in each mode so the distribution of the eigenvalues is expected to affect the performance of the algorithm. To study the MPCA properties on data of different characteristics, three synthetic data sets with eigenvalues in each mode spanning different magnitude ranges are generated. In particular, M third order tensor samples $\mathcal{A}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ are generated per set according to

$$\mathcal{A}_m = \mathcal{B}_m \times_1 \mathbf{C}^{(1)} \times_2 \mathbf{C}^{(2)} \times_3 \mathbf{C}^{(3)} + \mathcal{D}_m, \tag{4.27}$$

using a core tensor $\mathcal{B}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, n-mode projection matrix $\mathbf{C}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ $(n = 1, 2, 3)$ and a "noise" tensor $\mathcal{D}_m \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. All entries in $\mathcal{B}_m$ are drawn from a zero-mean unit-variance Gaussian distribution and are multiplied by $\left(\frac{I_1 \cdot I_2 \cdot I_3}{i_1 \cdot i_2 \cdot i_3}\right)^f$. In this data gen-

eration procedure, f controls the eigenvalue distributions, so that data sets are created having eigenvalues' magnitudes in different ranges. Smaller f results in a narrower range of eigenvalue spread. The matrices $\mathbf{C}^{(n)}$ $(n = 1, 2, 3)$ are orthogonal matrices obtained by applying SVD on random matrices with entries drawn from zero-mean, unit-variance Gaussian distribution. All entries of $\mathcal{D}_m$ are drawn from a zero-mean Gaussian distribution with variance 0.01. Three synthetic data sets, db1, db2, and db3, of size $30 \times 20 \times 10$ with $M = 100$ and $f = 1/2, 1/4$, and $1/16$, respectively, are created. Figure 4.5(a) depicts the spread of eigenvalue magnitudes and Fig. 4.5(b) depicts their eigenvalue cumulative distribution.

4.6.2 MPCA properties

First, the effects of the initial conditions are tested using the synthetic data sets. Both random matrices and pseudo identity matrices (truncated identity matrices) have been tested. Typical examples are shown in Fig. 4.6. From the simulation studies, it can be observed that despite the different initializations, the MPCA algorithm, when applied on sets db1 and db2, converges to the same point within three iterations. On set db3, the algorithm with different initializations converges to the same point within ten iterations for $Q \geq 0.35$. For small value of Q (< 0.35) on set db3, the algorithm using random matrices as initialization could converge to a point that is different from (lower than) the point to which the algorithm using the other two initialization methods converges, as shown in Fig. 4.6(c). This indicates that initialization methods could affect the final results on data sets with similar characteristics as db3 when a small Q is used. In summary, initialization has little effect on the final results for synthetic data sets db1 and db2 with all values of Q, and for synthetic data set db3 with $Q \geq 0.35$. In pattern recognition applications, it is often desired to keep most of the variation/energy in the original data and hence the proposed algorithm using different initializations is expected to converge well since $Q > 0.35$ is easily satisfied in practice. Since the MPCA

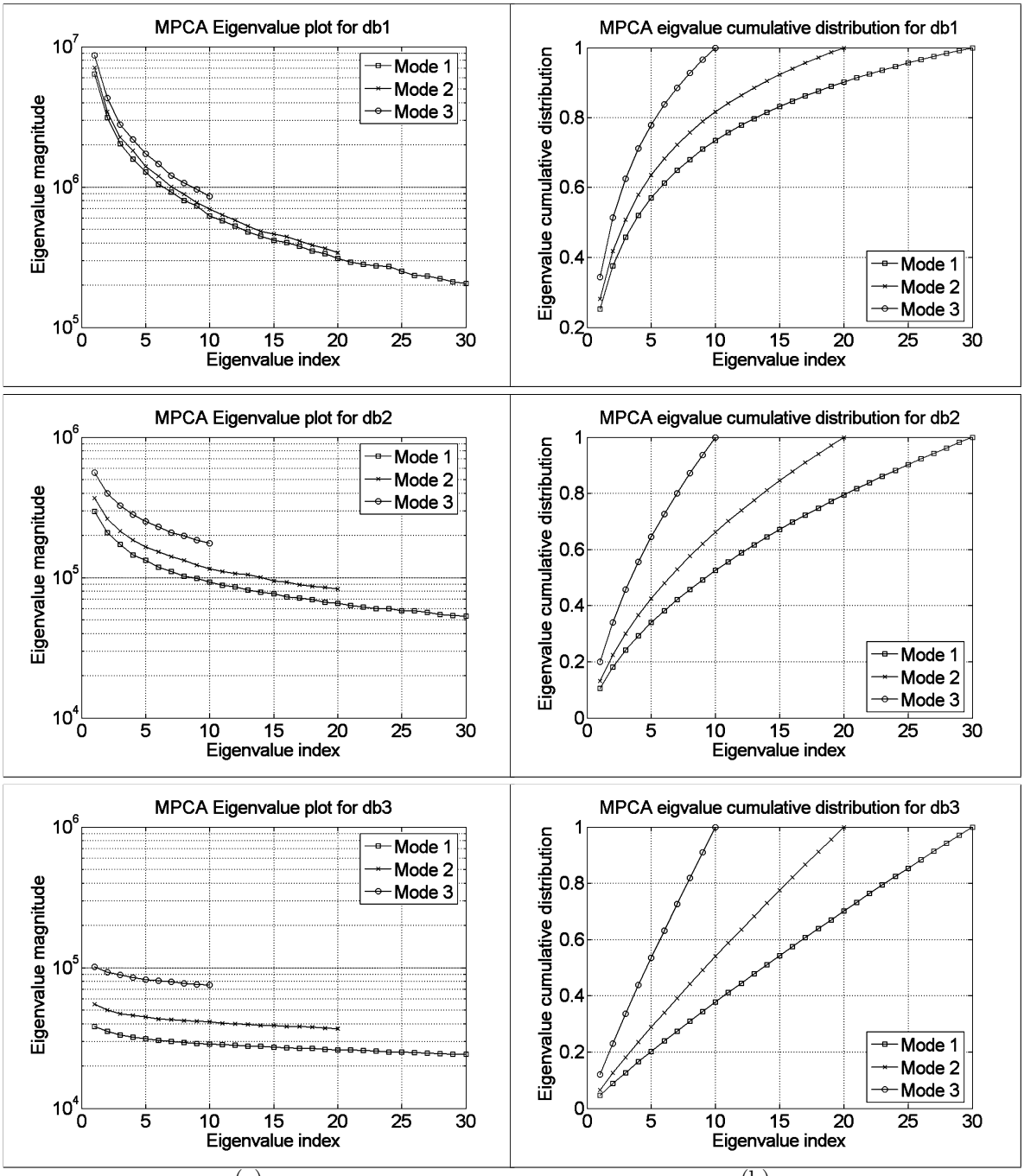


Figure 4.5: Plots of (a) the eigenvalue magnitudes, and (b) their cumulative distributions for the synthetic data sets: db1, db2 and db3.

algorithm using the proposed initialization, full projection truncation, converges faster than the algorithm using the other initialization methods, the full projection truncation is expected to be closer to the local maximum point and it is used for MPCA initialization

in this work.

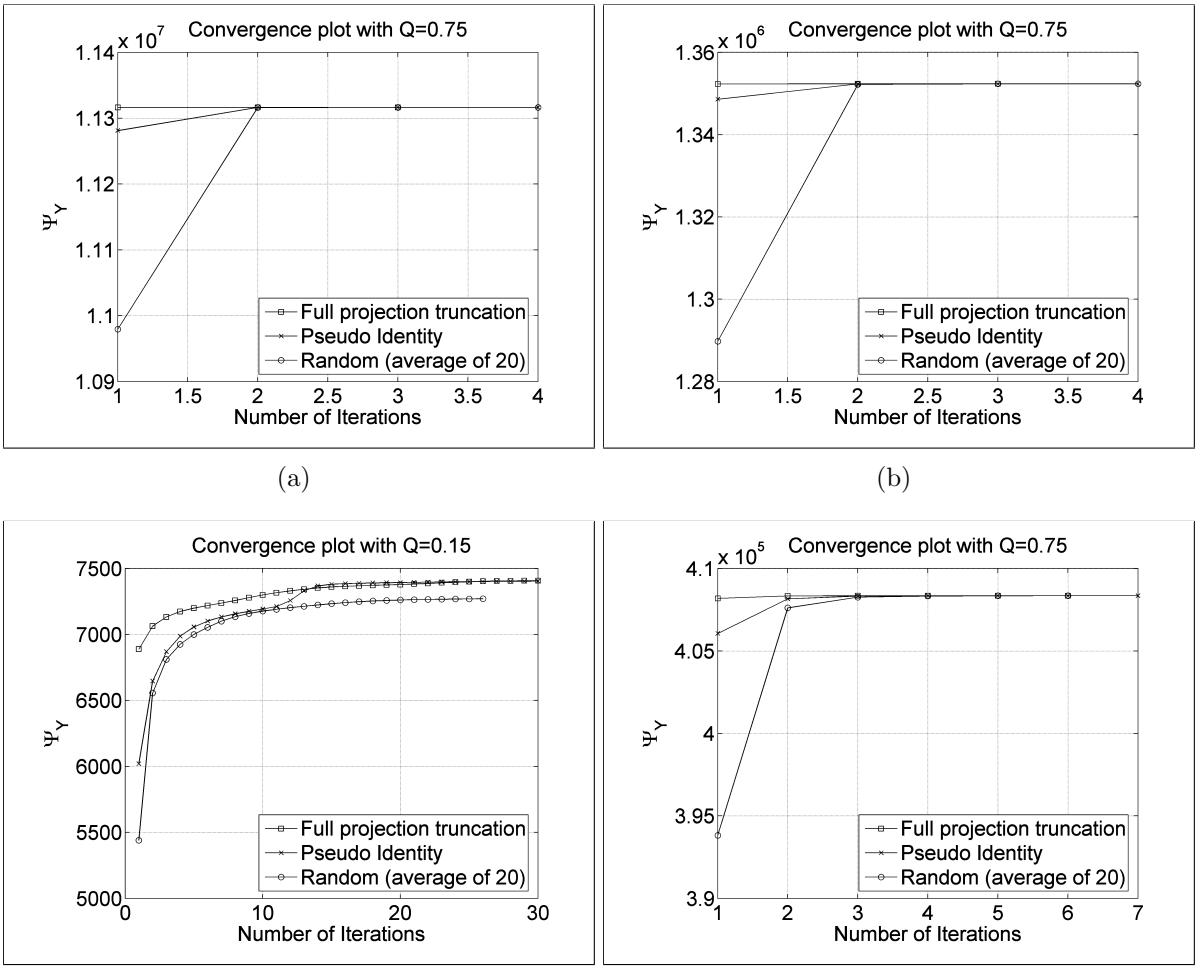
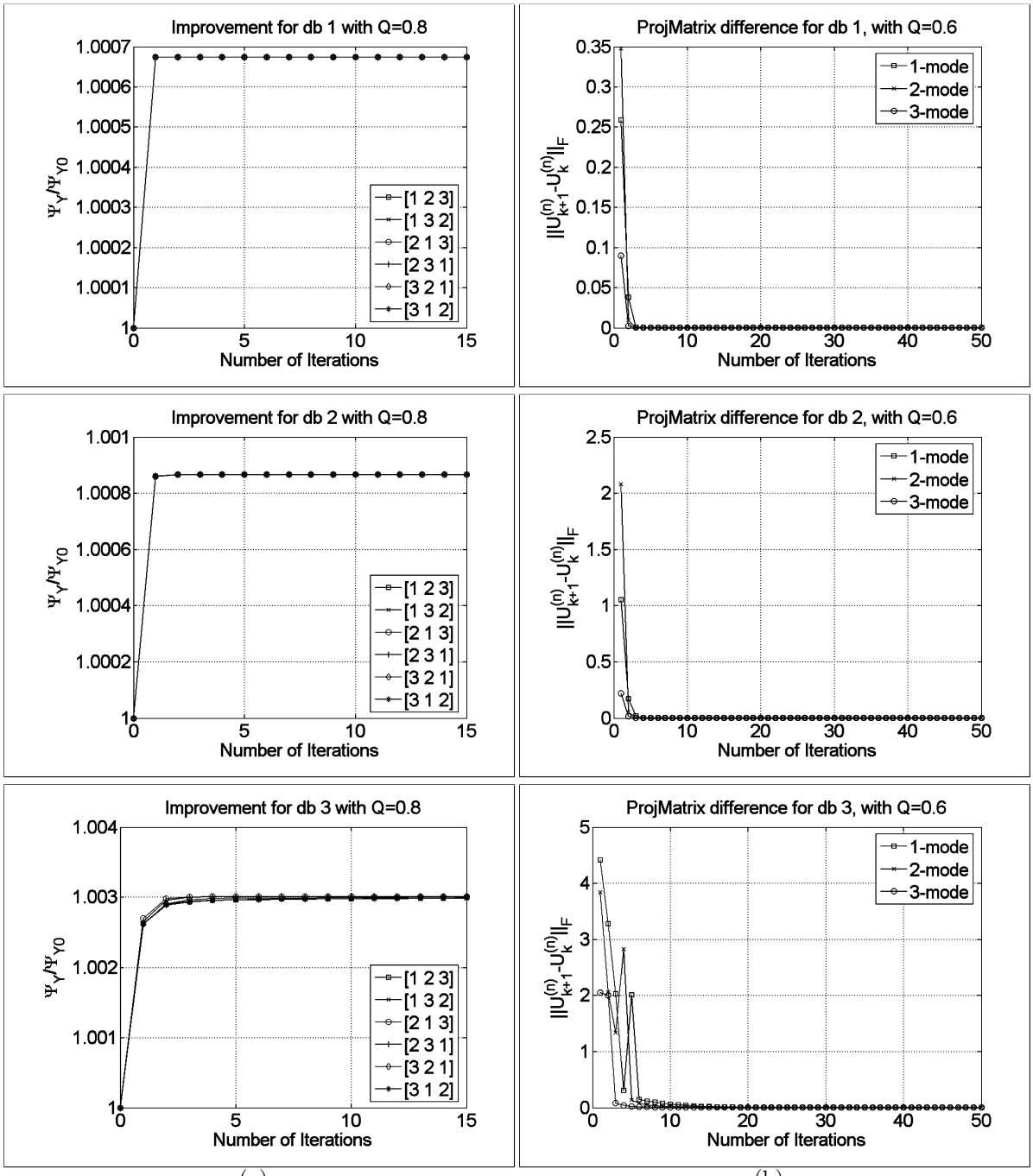


Figure 4.6: Convergence plots for MPCA with different initializations on (a) db1 with $Q = 0.75$, (b) db2 with $Q = 0.75$, (c) db3 with $Q = 0.15$, and (d) db3 with $Q = 0.75$.

Second, the effects of the projection order are studied. A typical example for $Q = 0.8$ is depicted in Fig. 4.7(a), which plots the ratio $\Psi_{\mathcal{Y}_k}/\Psi_{\mathcal{Y}_0}$ against the iteration number for up to 15 iterations with all six possible ordering. The simulation results indicate that there is no significant difference in the captured total scatter for db1 and db2, while there is some small difference for db3. For db3, the difference in total scatter captured using different orderings is negligible ($< 0.01\%$) for $Q > 0.5$ and it increases as Q decreases, e.g., the difference is about 1% when $Q = 0.2$. This observation is consistent with the poorer convergence property of db3, especially for a small Q, in other experiments such

as the initialization study above and the convergence study in the following.



(a) (b)

Figure 4.7: Illustration of (a) the effects of projection order with $Q = 0.8$, and (b) the convergence of projection matrices with $Q = 0.6$, in MPCA for the synthetic data sets: db1, db2 and db3.

To illustrate the convergence of the projection matrices, Fig. 4.7(b) shows some

typical results for $Q = 0.6$, where the evolution of successive projection matrix differences for the three modes are plotted against the number of iterations. From the figure, the projection matrices converge within five iterations for db1 and db2, while for db3, they converge within 15 iterations.

To study the evolution of the total scatter $\Psi_{\mathcal{Y}}$ over iterations, the ratio of the value of $\Psi_{\mathcal{Y}_k}$ over the initial value $\Psi_{\mathcal{Y}_0}$ is plotted against the number of iterations, as a function of dimensionality reduction determined by Q. For illustration purpose, results obtained for up to 15 iterations with $Q = 0.2$ and $Q = 0.8$ are shown in Figs. 4.8(a) and 4.8(b), respectively. As it can be seen from the figure, the first iteration results in the greatest increase in $\Psi_{\mathcal{Y}}$ while subsequent iterations result in smaller and smaller increments, especially for data sets db1 and db2. To study the empirical convergence performance, the number of iterations for convergence using a termination value of $\eta = 10^{-6}$ is plotted in Fig. 4.8(c) as a function of the parameter Q. These figures demonstrate that in MPCA, the number of iterations needed to converge decreases as the range spanned by the eigenvalues for the data samples or the value of Q increases.

The dependency on Q can be explained from two aspects. Q is closely related to the number of eigenvectors truncated. First, from Theorem 4.3, the bounds on $\Psi_{\mathcal{Y}_0}$ tend to become looser when the number of eigenvectors truncated increases (Q decreases), and vice versa. Looser (tighter) bounds tend to result in a poorer (better) initialization and it takes more (less) iterations to reach a local optimum. Second, by Theorem 4.2, more truncation (smaller value of Q) tends to decrease the eigenvalues in the other modes more so that more iterations are needed to converge, and vice versa. The dependency on the range of eigenvalue spread is less obvious. Narrower range in a mode means the variation along each eigenvector in this mode is similar and each of the truncated eigenvectors tends to encode similar amount of variation as each remaining one, which tends to make the convergence harder. On the other hand, broader range of the eigenvalue spread in a mode means that the energy is more concentrated in this mode and in this case,

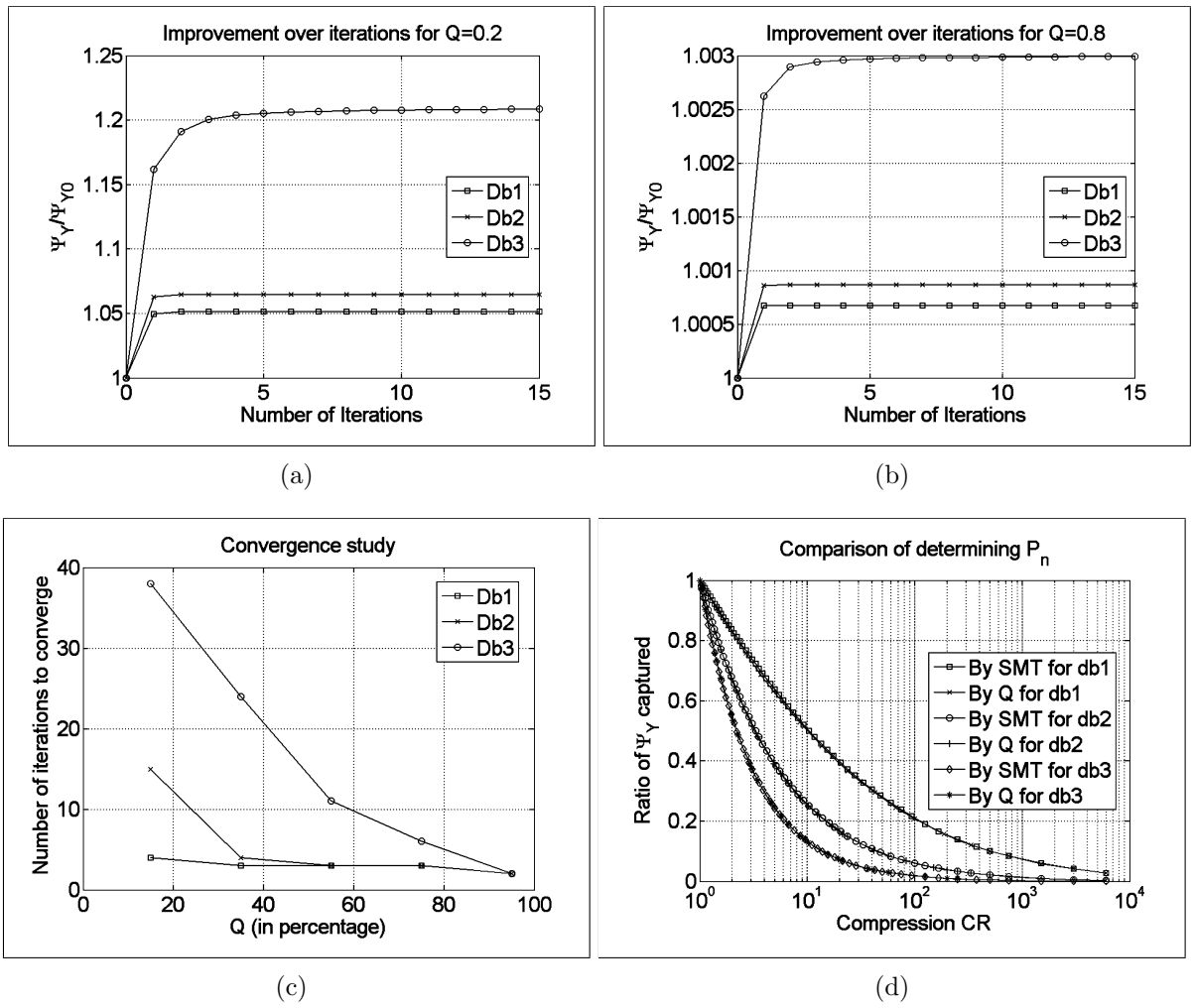


Figure 4.8: Illustration of various properties of MPCA on the synthetic data sets: (a) evolution of $\Psi_{\mathcal{Y}}$ for $Q = 0.2$, (b) evolution of $\Psi_{\mathcal{Y}}$ for $Q = 0.8$, (c) number of iterations to converge, and (d) SMT versus Q-based selection of P_n (SMT: sequential mode truncation).

convergence is expected to be easier.

In practical recognition tasks, Q is commonly set to a large value in order to capture most of the variation. Furthermore, the eigenvalues of practical data samples usually spread a wide range in each mode due to redundancy/correlation. For example, Figure 4.9 shows a plot of the eigenvalues in three modes and their cumulative distributions obtained from the gallery set of the USF gait database V.1.7, where the eigenvalues have a wide range in each mode. In particular, the 2-mode eigenvalues drop sharply after the

20th eigenvalue, indicating high redundancy in the 2-mode (row mode), and the 1-mode (column mode) eigenvalues decrease gradually except at the last point, which may be due to the centering. Therefore, in practice, the number of iterations K can be set to a small value such as one with little sacrifice in the variation captured while enjoying significant gain in empirical efficiency.

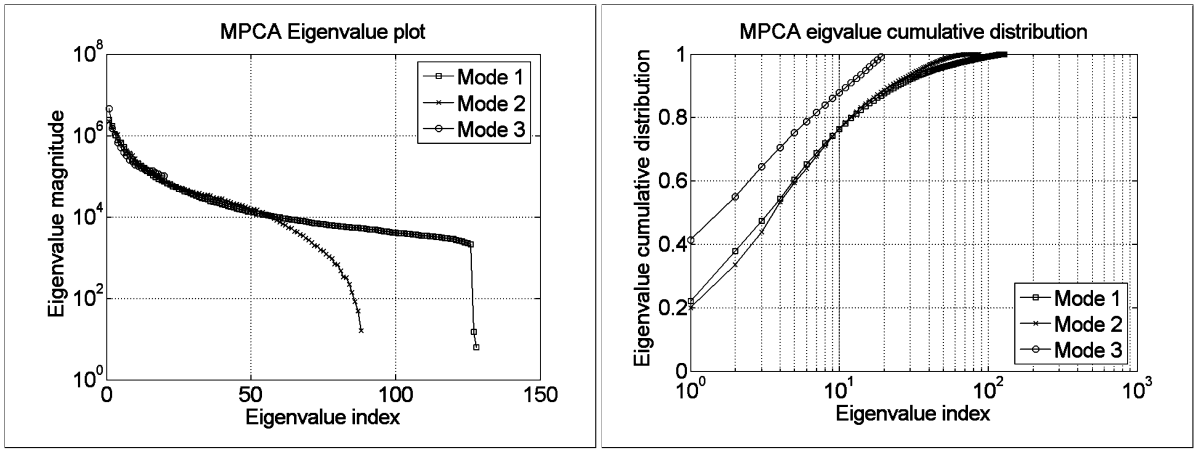


Figure 4.9: The eigenvalue magnitudes and their cumulative distributions for the gallery set of the USF gait database V.1.7.

Lastly, to investigate the Q-based method and the sequential mode truncation method for subspace dimensionality determination, P_n is determined for each mode n using these two methods with various degrees of truncation across all the modes. The resulted total scatters are compared under the same compression ratio CR (the same reduced dimensionality), as shown in Fig. 4.8(d). The figure indicates that these two methods have very similar results for all the three data sets. Hence, in practice, the more efficient Q-based method is used to determine the tensor subspace dimensionality. In addition, it is also verified through experiments that truncation in all modes is advantageous against truncation in only one or two modes.

4.7 Summary

This chapter has proposed the MPCA algorithm as a multilinear extension of PCA for general tensor objects. MPCA determines a tensor-to-tensor projection onto a tensor subspace of lower dimensionality that captures most of the signal variation present in the original tensorial representation. An iterative solution has been formulated and issues including initialization, projection order, convergence, and subspace dimensionality determination have been discussed in detail. Then, by viewing the MPCA projection as a number of eigentensors, a scheme of discriminative eigentensor selection has been proposed for better recognition. The MPCA+LDA approach has also been introduced. Furthermore, the combination of MPCA with the ensemble-based learning technology named boosting has been investigated. The proposed MPCA+boosting method not only results in more efficient processing, but also offers one more weakness control mechanism, which is important in the LDA-based booster. In the experimental section, synthetic data sets have been constructed and various MPCA properties have been studied. The experimental study helps better understanding of MPCA and offers practical recommendations for its algorithm settings, such as initialization, projection order, termination, and subspace dimensionality determination.

Nonetheless, in the development of MPCA, the correlation among features is not considered and the MPCA features obtained may not have zero correlations as in the classical PCA. The next chapter continues to explore the unsupervised multilinear subspace learning by examining uncorrelated feature extraction within the PCA paradigm, but in a multilinear setting.

Chapter 5

Uncorrelated Multilinear Principal Component Analysis

5.1 Introduction

As pointed out in Section 3.4.2 (page 54), none of the existing unsupervised multilinear subspace learning algorithms takes an important property of the classical PCA into account, i.e., PCA derives uncorrelated features, which contain minimum redundancy and ensure linear independence among features. Instead, most of them produce orthogonal bases in each mode, and the correlations among features obtained are not guaranteed to be zero in a multilinear setting, unlike in PCA. For the approach of unsupervised multilinear subspace learning through the tensor-to-tensor projection, it is extremely difficult, if not impossible, to enforce the zero-correlation constraint. As analyzed in Section 2.2.4 (page 25), the tensor-to-tensor projection effectively consists of a number of interdependent elementary multilinear projections and the respective features obtained are likely to be correlated rather than uncorrelated.

On the other hand, it is noted in the discussions in Section 3.5 (page 60) that in unsupervised multilinear subspace learning through the tensor-to-vector projection, the

approach of variation maximization has not been studied. The only method in this category tries to minimize the reconstruction error through a heuristic approach.

Motivated by the above discussions, this chapter investigates the development of the uncorrelated multilinear PCA (UMPCA), an unsupervised multilinear subspace learning algorithm that produces uncorrelated features through the tensor-to-vector projection. Accordingly, this chapter addresses the shaded empty box in Fig. 3.7(a) (page 51) under the tensor-to-vector projection. The derivation of the UMPCA algorithm follows the classical PCA derivation of successive variance maximization [47]. A number of elementary multilinear projections are solved one by one to maximize the captured variance with the zero-correlation constraint enforced. As in MPCA, the solution is iterative in nature and it solves the multilinear problem through a series of linear problems. Issues related to initialization, projection order, termination, and convergence are studied and its relationships with existing solutions are discussed. In addition, a theoretical justification is given on a limitation of UMPCA in the number of uncorrelated features that can be extracted, and ways to work around this limitation are suggested.

The rest of this chapter is organized as follows. In Section 5.2, the problem of UMPCA is first formulated. The solution is then derived as a sequential iterative process. Next, in this section, relationships with existing algorithms and its limitation are analyzed. Issues on initialization, projection order, termination, and convergence are also discussed, followed by the computational aspects. Section 5.3 presents the experimental study of the UMPCA properties on the synthetic data sets used in Section 4.6 (page 86). Finally, this chapter is summarized in Section 5.4.

5.2 The UMPCA Algorithm

This section first formulates the UMPCA objective function and then adopts the classical successive variance maximization approach in the derivation of PCA together with the

alternating projection method as in MPCA to solve the problem. Thereafter, connections with existing methods, and the design and computational issues are discussed.

In the presentation of UMPCA, for the convenience of discussion, the training samples are assumed to be zero-mean. When the training sample mean is not zero, it can be subtracted to make the training samples to be zero-mean. In this case, the constraint of uncorrelated features is the same as orthogonal features, as shown in the following proposition.

Proposition 5.1. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be vector observations of the variables* x *and* y*. Then,* $\mathbf{x}$ *and* $\mathbf{y}$ *are orthogonal iff* $\mathbf{x}^T\mathbf{y} = 0$*, and* $\mathbf{x}$ *and* $\mathbf{y}$ *are uncorrelated iff* $(\mathbf{x} - \bar{x})^T(\mathbf{y} - \bar{y}) = 0$*, where* $\bar{x}$ *and* $\bar{y}$ *are the means of* $\mathbf{x}$ *and* $\mathbf{y}$*, respectively [103]. Thus, two zero-mean (*$\bar{x} = \bar{y} = 0$*) vectors are uncorrelated when they are orthogonal [54].*

5.2.1 The UMPCA problem

Following the standard derivation of PCA given in [47], the variance of the principal components is considered one by one. In the setting of the tensor-to-vector projection, the pth principal components are $\{y_{m_p}, m = 1, ..., M\}$, where M is the number of training samples and y_{m_p} is the projection of the mth sample $\mathcal{X}_m$ by the pth elementary multilinear projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$: $y_{m_p} = \mathcal{X}_m \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$. Accordingly, from Definition 2.4 (page 32), the variance is measure by their total scatter $S_{T_p}^{\mathbf{y}}$:

$$S_{T_p}^{\mathbf{y}} = \sum_{m=1}^{M} (y_{m_p} - \bar{y}_p)^2, \tag{5.1}$$

where $\bar{y}_p = \frac{1}{M}\sum_m y_{m_p}$. In addition, let $\mathbf{g}_p$ denote the pth coordinate vector, with its mth component $\mathbf{g}_p(m) = y_{m_p}$.

A formal definition of the UMPCA problem is then given in the following:

A set of M tensor object samples $\{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_M\}$ (with zero-mean) is available for training. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 ... \times I_N}$ assumes values in the tensor space

$\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$. The objective of UMPCA is to find a tensor-to-vector projection, which consists of P elementary multilinear projections, $\{\mathbf{u}_p^{(n)} \in \mathbb{R}^{I_n \times 1}, n = 1, ..., N\}_{p=1}^{P}$ that maps 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 $P < \prod_{n=1}^{N} I_n$):

$$\mathbf{y}_m = \mathcal{X}_m \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}, m = 1, ..., M, \tag{5.2}$$

such that the variance of the projected samples, measured by $S_{T_p}^{\mathbf{y}}$, is maximized in each elementary multilinear projection direction, subject to the constraint that the P coordinate vectors $\{\mathbf{g}_p \in \mathbb{R}^M, p = 1, ..., P\}$ are uncorrelated.

In other words, the UMPCA objective is to determine a set of P elementary multilinear projections $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ that maximize the variance while producing features with zero-correlation. Thus, the objective function for the pth elementary multilinear projection is

$$\begin{aligned} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\} &= \arg\max \sum_{m=1}^{M} (y_{m_p} - \overline{y}_p)^2, \\ \texttt{subject to } \mathbf{u}_p^{(n)^T} \mathbf{u}_p^{(n)} = 1 \quad &\text{and} \quad \frac{\mathbf{g}_p^T \mathbf{g}_q}{\| \mathbf{g}_p \| \| \mathbf{g}_q \|} = \delta_{pq},\ p, q = 1, ..., P, \end{aligned} \tag{5.3}$$

where δ_{pq} is the Kronecker delta defined as

$$\delta_{pq} = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{Otherwise.} \end{cases} \tag{5.4}$$

Remark 5.1. *It should be noted that despite working directly on tensorial data, the UMPCA algorithm is a feature extraction algorithm that produces feature vectors like traditional linear algorithms, due to the nature of the tensor-to-vector projection. For a test sample $\mathcal{X}$, the feature vector $\mathbf{y}$ is obtained through the tensor-to-vector projection*

obtained by UMPCA as:

$$\mathbf{y} = \mathcal{X} \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}. \tag{5.5}$$

5.2.2 The derivation of UMPCA

To solve this UMPCA problem (5.3), the successive variance maximization approach in the derivation of PCA in [47] is followed. The P elementary multilinear projections $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ are determined sequentially (one by one) in P steps, with the pth step obtaining the pth elementary multilinear projection. This stepwise process proceeds as:

Step 1**:** Determine the first elementary multilinear projection $\{\mathbf{u}_1^{(n)^T}, n = 1, ..., N\}$ by maximizing $S_{T_1}^{\mathbf{y}}$ without any constraint.

Step 2**:** Determine the second elementary multilinear projection $\{\mathbf{u}_2^{(n)^T}, n = 1, ..., N\}$ by maximizing $S_{T_2}^{\mathbf{y}}$ subject to the constraint that $\mathbf{g}_2^T \mathbf{g}_1 = 0$.

Step 3**:** Determine the third elementary multilinear projection $\{\mathbf{u}_3^{(n)^T}, n = 1, ..., N\}$ by maximizing $S_{T_3}^{\mathbf{y}}$ subject to the constraint that $\mathbf{g}_3^T \mathbf{g}_1 = 0$ and $\mathbf{g}_3^T \mathbf{g}_2 = 0$.

Step p $(p = 4, ..., P)$**:** Determine the pth elementary multilinear projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$ by maximizing $S_{T_p}^{\mathbf{y}}$ subject to the constraint that $\mathbf{g}_p^T \mathbf{g}_q = 0$ for $q = 1, ..., p-1$.

In the following, the algorithm to compute these elementary multilinear projections is presented in detail, as summarized in the pseudo-code in Fig. 5.1. In the figure, the stepwise process described above corresponds to the loop indexed by p.

In order to solve for the pth elementary multilinear projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$, there are N sets of parameters corresponding to the N projection vectors to be determined, $\mathbf{u}_p^{(1)}, \mathbf{u}_p^{(2)}, ...\mathbf{u}_p^{(N)}$, one in each mode. It will be desirable to determine these N sets

Input: A set of tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$, the desired feature vector length P, the maximum number of iterations K, and a small number η for testing convergence.

Output: The P elementary multilinear projections $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ that captures the most variance in the projected space.

Algorithm:

For $p = 1 : P$ **(step** p**: determine the** p**th elementary multilinear projections)**

If $p > 1$, calculate the coordinate vector $\mathbf{g}_{p-1}$: $\mathbf{g}_{p-1}(m) = \mathcal{X}_m \times_1 \mathbf{u}_{p-1}^{(1)^T} \times_2 \mathbf{u}_{p-1}^{(2)^T} ... \times_N \mathbf{u}_{p-1}^{(N)^T}$.

- For $n = 1, ..., N$, initialize $\mathbf{u}_{p_{(0)}}^{(n)} \in \mathbb{R}^{I_n}$.
- For $k = 1 : K$
  - For $n = 1 : N$
    - Calculate $\tilde{\mathbf{y}}_{m_p}^{(n)} = \mathcal{X}_m \times_1 \mathbf{u}_{p_{(k)}}^{(1)^T} ... \times_{n-1} \mathbf{u}_{p_{(k)}}^{(n-1)^T} \times_{n+1} \mathbf{u}_{p_{(k-1)}}^{(n+1)^T} ... \times_N \mathbf{u}_{p_{(k-1)}}^{(N)^T}$, for $m = 1, ..., M$.
    - Calculate $\mathbf{\Upsilon}_p^{(n)}$ and $\tilde{\mathbf{S}}_{T_p}^{(n)}$. Set $\mathbf{u}_{p_{(k)}}^{(n)}$ to be the (unit) eigenvector of $\mathbf{\Upsilon}_p^{(n)} \tilde{\mathbf{S}}_{T_p}^{(n)}$ associated with the largest eigenvalue.
  - If $k = K$ or $(S_{T_{p_k}}^{\mathbf{y}} - S_{T_{p_{(k-1)}}}^{\mathbf{y}})/S_{T_{p_{(k-1)}}}^{\mathbf{y}} < \eta$, set $\mathbf{u}_p^{(n)} = \mathbf{u}_{p_k}^{(n)}$ for all n, break.
- **Output** $\{\mathbf{u}_p^{(n)}\}$. Go the step $p + 1$ if $p < P$. Stop if $p = P$.

Figure 5.1: The pseudo-code implementation of the UMPCA algorithm for feature extraction from tensor objects.

of parameters (N projection vectors) in all modes simultaneously so that $S_{T_p}^{\mathbf{y}}$ is (globally) maximized, subject to the zero-correlation constraint. Unfortunately, as in MPCA, this is a rather complicated nonlinear problem without an existing optimal solution, except when $N = 1$, which is the classical PCA where only one projection vector is to be solved. Therefore, the alternating projection method in the ALS algorithm [12, 35, 56] is used

again to solve this multilinear problem, where a multilinear optimization problem is reduced into smaller conditional subproblems that can be solved through simple established methods employed in the linear case.

For each elementary multilinear projection to be determined, the parameters of the projection vector $\mathbf{u}_p^{(n^*)}$ for each mode n^* are estimated one by one separately, conditioned on $\{\mathbf{u}_p^{(n)}, n \neq n^*\}$, the parameter values of the projection vectors for the other modes. Thus, by fixing $\{\mathbf{u}_p^{(n)}, n \neq n^*\}$, a new objective function depending only on $\mathbf{u}_p^{(n^*)}$ is formulated. This conditional subproblem is linear and much simpler. The parameter estimations for each mode are obtained in this way sequentially and iteratively until a stopping criterion is met. It corresponds to the loop indexed by k in Fig. 5.1, and in each iteration k, the loop indexed by n in Fig. 5.1 consists of the N conditional subproblems.

To solve for $\mathbf{u}_p^{(n^*)}$ in the n^*-mode, assuming that $\{\mathbf{u}_p^{(n)}, n \neq n^*\}$ is given, the tensor samples are projected in these $(N-1)$ modes $\{n \neq n^*\}$ first to obtain the vectors

$$\tilde{\mathbf{y}}_{m_p}^{(n^*)} = \mathcal{X}_m \times_1 \mathbf{u}_p^{(1)^T} ... \times_{n^*-1} \mathbf{u}_p^{(n^*-1)^T} \times_{n^*+1} \mathbf{u}_p^{(n^*+1)^T} ... \times_N \mathbf{u}_p^{(N)^T}, \tag{5.6}$$

where $\tilde{\mathbf{y}}_{m_p}^{(n^*)} \in \mathbb{R}^{I_{n^*}}$. This conditional subproblem then becomes to determine $\mathbf{u}_p^{(n^*)}$ that projects the vector samples $\{\tilde{\mathbf{y}}_{m_p}^{(n^*)}, m = 1, ..., M\}$ onto a line so that the variance is maximized, subject to the zero-correlation constraint, which is a PCA problem with the input samples $\{\tilde{\mathbf{y}}_{m_p}^{(n^*)}, m = 1, ..., M\}$. The corresponding total scatter matrix $\tilde{\mathbf{S}}_{T_p}^{(n^*)}$ is then defined as

$$\tilde{\mathbf{S}}_{T_p}^{(n^*)} = \sum_{m=1}^{M} (\tilde{\mathbf{y}}_{m_p}^{(n^*)} - \bar{\tilde{\mathbf{y}}}_p^{(n^*)})(\tilde{\mathbf{y}}_{m_p}^{(n^*)} - \bar{\tilde{\mathbf{y}}}_p^{(n^*)})^T, \tag{5.7}$$

where $\bar{\tilde{\mathbf{y}}}_p^{(n^*)} = \frac{1}{M} \sum_m \tilde{\mathbf{y}}_{m_p}^{(n^*)}$. With (5.7), the P elementary multilinear projections can be solved sequentially. For $p = 1$, the $\mathbf{u}_1^{(n^*)}$ that maximizes the total scatter $\mathbf{u}_1^{(n^*)^T} \tilde{\mathbf{S}}_{T_1}^{(n^*)} \mathbf{u}_1^{(n^*)}$ in the projected space is obtained as the unit eigenvector of $\tilde{\mathbf{S}}_{T_1}^{(n^*)}$ associated with the largest eigenvalue. Next, the pth $(p > 1)$ elementary multilinear projection is determined.

Given the first $(p-1)$ elementary multilinear projections, the pth elementary multilinear projection aims to maximize the total scatter $S_{T_p}^{\mathbf{y}}$, subject to the constraint that features projected by the pth elementary multilinear projection are uncorrelated with those projected by the first $(p-1)$ elementary multilinear projections. Let $\tilde{\mathbf{Y}}_p^{(n^*)} \in \mathbb{R}^{I_{n^*} \times M}$ be a matrix with $\tilde{\mathbf{y}}_{m_p}^{(n^*)}$ as its mth column, i.e.,

$$\tilde{\mathbf{Y}}_p^{(n^*)} = \left[\tilde{\mathbf{y}}_{1_p}^{(n^*)}, \tilde{\mathbf{y}}_{2_p}^{(n^*)}, ..., \tilde{\mathbf{y}}_{M_p}^{(n^*)}\right], \tag{5.8}$$

then the pth coordinate vector is $\mathbf{g}_p = \tilde{\mathbf{Y}}_p^{(n^*)^T} \mathbf{u}_p^{(n^*)}$. The constraint that $\mathbf{g}_p$ is uncorrelated with $\{\mathbf{g}_q, q = 1, ..., p-1\}$ can be written as

$$\mathbf{g}_p^T \mathbf{g}_q = \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = 0, q = 1, ..., p-1. \tag{5.9}$$

Thus, $\mathbf{u}_p^{(n^*)}$ $(p > 1)$ can be determined by solving the following constrained optimization problem:

$$\mathbf{u}_p^{(n^*)} = \arg\max \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u}_p^{(n^*)}, \tag{5.10}$$

$$\texttt{subject to } \mathbf{u}_p^{(n^*)^T} \mathbf{u}_p^{(n^*)} = 1 \text{ and } \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = 0, q = 1, ..., p-1, \tag{5.11}$$

The solution is given by the following theorem:

Theorem 5.1. *The solution to the problem (5.10) is the (unit-length) eigenvector corresponding to the largest eigenvalue of the following eigenvalue problem:*

$$\boldsymbol{\Upsilon}_p^{(n^*)} \tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u} = \lambda \mathbf{u}, \tag{5.12}$$

where

$$\boldsymbol{\Upsilon}_p^{(n^*)} = \mathbf{I}_{I_{n^*}} - \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \boldsymbol{\Gamma}_p^{-1} \mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T}, \tag{5.13}$$

$$\mathbf{\Gamma}_p = \mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1}, \tag{5.14}$$

$$\mathbf{G}_{p-1} = [\mathbf{g}_1 \quad \mathbf{g}_2 \quad ... \mathbf{g}_{p-1}] \in \mathbb{R}^{M \times (p-1)}, \tag{5.15}$$

and $\mathbf{I}_{I_{n^*}}$ *is an identity matrix of size* $I_{n^*} \times I_{n^*}$.

Proof. The proof of Theorem 5.1 is given in Appendix A.5. □

By setting $\mathbf{\Upsilon}_1^{(n^*)} = \mathbf{I}_{I_{n^*}}$ and from Theorem 5.1, a unified solution for UMPCA is obtained: for $p = 1, ..., P$, $\mathbf{u}_p^{(n^*)}$ is obtained as the unit eigenvector of $\mathbf{\Upsilon}_p^{(n^*)} \tilde{\mathbf{S}}_{T_p}^{(n^*)}$ associated with the largest eigenvalue.

It should be noted that this algorithm has a limitation in the number of uncorrelated features that can be extracted, given by the following corollary.

Corollary 5.1. *The number of uncorrelated features that can be extracted by UMPCA,* P*, is upper-bounded by* $\min\{\min_n I_n, M\}$*, i.e.,* $P \leq \min\{\min_n I_n, M\}$*, provided that the elements of* $\tilde{\mathbf{Y}}_p^{(n)}$ *are not all zero.*

Proof. The proof of Corollary 5.1 is given in Appendix A.6. □

The conclusion in Corollary 5.1 is expected because the elementary multilinear projections to be solved in UMPCA correspond to highly constrained situations in the linear case where the features extracted are constrained by both their correlation property and the simplicity of the projection. This implies that UMPCA may be more suitable for high resolution tensor objects where the dimensionality in each mode is high enough to enable the extraction of sufficient number of (uncorrelated) features. UMPCA is also useful for applications that need only a small number of features, such as clustering of a small number of classes. On the other hand, the UMPCA features may be combined with other features such as those extracted by PCA, TROD, and MPCA when there are more features needed. In this case, either the zero-correlation constraint (for MPCA and TROD) or the constraint on the simplicity of the projection (for PCA) has to be relaxed.

5.2.3 Connections to existing solutions

The UMPCA algorithm follows the approach of successive variance maximization in the classical derivation of PCA [47]. Hence, when $N = 1$, the samples are vectors $\{\mathbf{x}_m \in \mathbb{R}^{I_1}\}$ with only one mode and UMPCA reduces to PCA. This can also be seen from Section 2.2.4 (page 25). Accordingly, in each step p, there is only one projection vector $\mathbf{u}_p$ to be solved to maximize the captured variance, subject to the zero-correlation constraint. Also, Corollary 5.1 indicates the maximum number of extracted features does not exceed $\min\{I_1, M\}$, which can be verified in PCA as the bound for the rank of the data matrix.

From Section 3.4.2 (page 54), TROD [116] shares similarity with UMPCA in that it also seeks for a tensor-to-vector projection to optimize an objective function. However, TROD minimizes the reconstruction error instead, and it takes a heuristic greedy approach through residue calculation while UMPCA takes a systematic, more principled formulation by taking consideration of the correlations among features.

5.2.4 Initialization, projection order, termination, and convergence

In this subsection, design issues of UMPCA, including the initialization method, the projection order, the termination conditions, and the convergence issue, are discussed.

As the determination of each elementary multilinear projection $\{\mathbf{u}_p^{(n)}, n = 1, ..., N\}$ is an iterative procedure due to the multilinear nature of UMPCA, initial estimations for the projection vectors $\{\mathbf{u}_p^{(n)}\}$ are necessary. However, there is no guidance from either the algorithm or the data on the best initialization that could result in the most variance captured. Thus, the determination of the optimal initialization in UMPCA is still an open problem, as in most iterative algorithms including other multilinear learning algorithms [150, 124, 142, 123]. In this dissertation, an empirically study is presented on two simple and commonly used initialization methods [86]: uniform initialization

and random initialization [142, 123], which do not depend on the data. In the uniform initialization, all n-mode projection vectors are initialized to have unit length and the same value along the I_n dimensions in n-mode, equivalent to the all ones vector $\mathbf{1}$ with proper normalization. In random initialization, each element of the n-mode projection vectors is drawn randomly from a zero-mean uniform distribution between $[-0.5, 0.5]$ and the initialized projection vectors are normalized to have unit length. The empirical studies in Section 5.3 indicate that the results of UMPCA are affected by initialization, and the uniform initialization gives more stable results.

The mode ordering (the inner-most "for loop" in Fig. 5.1, indexed by n) in computing the projection vectors affects the solution as well. Similar to initialization, there is no way to determine the optimal projection order and it is considered to be an open problem too. Empirical studies on the effects of the projection order indicate that with all the other algorithm settings fixed, altering the projection order does result in differences in the variance captured, but there is no guidance from either the data or the algorithm on what projection order is the best in the iteration. Therefore, there is no preference on a particular projection order. In practice, the projection vectors are solved sequentially (from 1-mode to N-mode), as in MPCA.

As seen from Fig. 5.1, the iterative procedure terminates if $(S^{\mathbf{y}}_{T_{p_k}} - S^{\mathbf{y}}_{T_{p_{(k-1)}}})/S^{\mathbf{y}}_{T_{p_{(k-1)}}} < \eta$, where $S^{\mathbf{y}}_{T_{p_k}}$ is the total scatter captured by the pth elementary multilinear projection obtained in the kth iteration of UMPCA and η is a small number threshold. Alternatively, the convergence of the projection vectors can also be examined: $dist\left(\mathbf{u}^{(n)}_{p_{(k)}}, \mathbf{u}^{(n)}_{p_{(k-1)}}\right) < \epsilon$, where ϵ is a user-defined small number threshold (e.g., $\epsilon = 10^{-3}$). This distance is defined as

$$dist\left(\mathbf{u}^{(n)}_{p_{(k)}}, \mathbf{u}^{(n)}_{p_{(k-1)}}\right) = \min\left(\| \mathbf{u}^{(n)}_{p_{(k)}} + \mathbf{u}^{(n)}_{p_{(k-1)}} \|, \| \mathbf{u}^{(n)}_{p_{(k)}} - \mathbf{u}^{(n)}_{p_{(k-1)}} \|\right) \tag{5.16}$$

since eigenvectors are unique up to sign. As to be shown in Section 5.3, the variance captured by a particular elementary multilinear projection usually increases rapidly for the first a few iterations and slowly afterwards. Therefore, the iteration can be termi-

nated by simply setting a maximum number of iterations K in practice for convenience, especially when the computational cost is a concern.

Regarding convergence, the derivation of Theorem 5.1 (Appendix A.5) implies that per iteration, the scatter $S_{T_p}^{\mathbf{y}}$ is a non-decreasing function (it either remains the same or increases) since each update of the projection vector $\mathbf{u}_p^{(n^*)}$ in a given mode n^* maximizes $S_{T_p}^{\mathbf{y}}$. On the other hand, $S_{T_p}^{\mathbf{y}}$ is upper-bounded by the variation in the original samples. Therefore, UMPCA is expected to convergence over iterations. Empirical results presented in Section 5.3 indicate that the proposed UMPCA algorithm converges within 10 iterations for typical tensor objects. Furthermore, when the largest eigenvalues in each mode are with multiplicity 1, which is the case for the simulated data and the face and gait data, the projection vectors $\{\mathbf{u}_p^{(n)}\}$, which maximize the objective function $S_{T_p}^{\mathbf{y}}$, are expected to converge as well, where the convergence is up to sign. Simulation studies show that the projection vectors $\{\mathbf{u}_p^{(n)}\}$ do converge over a number of iterations.

5.2.5 Computational aspects of UMPCA

Finally, the computation aspects of UMPCA is considered here. Specifically, the computational complexity and memory requirements are analyzed, in a similar way as in Section 4.3.7 (page 76) for MPCA. In the analysis, it is assumed again that $I_1 = I_2 = ... = I_N = \left(\prod_{n=1}^{N} I_n\right)^{\frac{1}{n}} = I$ for simplicity.

The most computational demanding steps in UMPCA are the calculations of the projection $\tilde{\mathbf{y}}_{m_p}^{(n)}$, the computation of $\tilde{\mathbf{S}}_{T_p}^{(n)}$ and $\boldsymbol{\Upsilon}_p^{(n)}$, and the calculation of the leading eigenvector of $\boldsymbol{\Upsilon}_p^{(n)}\tilde{\mathbf{S}}_{T_p}^{(n)}$. The complexity of calculating $\tilde{\mathbf{y}}_{m_p}^{(n)}$ for $m = 1, ..., M$ and $\tilde{\mathbf{S}}_{T_p}^{(n)}$ are in order of $O(M \cdot \sum_{n=2}^{N} I^n)$ and $O(M \cdot I^2)$, respectively. The computation of $\boldsymbol{\Upsilon}_p^{(n)}$ is in order of

$$\begin{aligned} & O\left(I \cdot M \cdot (p-1) + (p-1) \cdot I \cdot (p-1) + (p-1)^3 + I \cdot (p-1)^2 + I \cdot (p-1) \cdot I\right) \\ = \; & O\left((p-1) \cdot \left[I \cdot M + 2 \cdot I \cdot (p-1) + (p-1)^2 + I^2\right]\right). \end{aligned} \tag{5.17}$$

Lastly, the computation of $\mathbf{\Upsilon}_p^{(n)}\tilde{\mathbf{S}}_{T_p}^{(n)}$ and its eigen-decomposition[1] are both of order $O(I^3)$. Therefore, the computational complexity per mode n for one iteration k of step p is

$$O\left(M\left[I^2+\sum_{n=2}^{N}I^n\right]+(p-1)\left[I\cdot M+2I(p-1)+(p-1)^2+I^2\right]+2I^3\right). \quad (5.18)$$

Regarding the memory requirement, as in MPCA, the respective computation can be done incrementally by reading $\mathcal{X}_m$ sequentially. Thus, except for $N=1$, the memory needed for the UMPCA algorithm can be as low as $O(I^N)$, although sequential reading will lead to higher I/O cost.

From the analysis above, it can be seen that as a sequential iterative solution, UMPCA may have a high computational and I/O cost. Nonetheless, since solving the UMPCA projection is only in the training phase of the targeted recognition tasks, it can be done offline and the additional computational and I/O cost due to iterations and sequential processing are not considered a disadvantage. In the testing phase, the extraction (projection) of features from a test sample is a linear operation, as efficient as conventional linear subspace algorithms.

5.3 Experimental Study

This section investigates the various properties of the UMPCA algorithm. The face and gait recognition performance and the comparisons against existing solutions will be examined in detail in Chapter 7 as well. The UMPCA properties studied here are: a) the effects of initialization, b) the effects of projection order, and c) the convergence of the algorithm. Since UMPCA and MPCA are both unsupervised algorithms with the objective of variance maximization, the properties of UMPCA are similarly affected by the eigenvalue distribution of the input data. For the same reason as in the MPCA

[1]Since only the largest eigenvalue and the corresponding eigenvector is needed in UMPCA, more efficient computational methods may be applied in practice.

property study, the section reports only the experimental results on the three synthetic data sets generated in Section 4.6.1 (page 86).

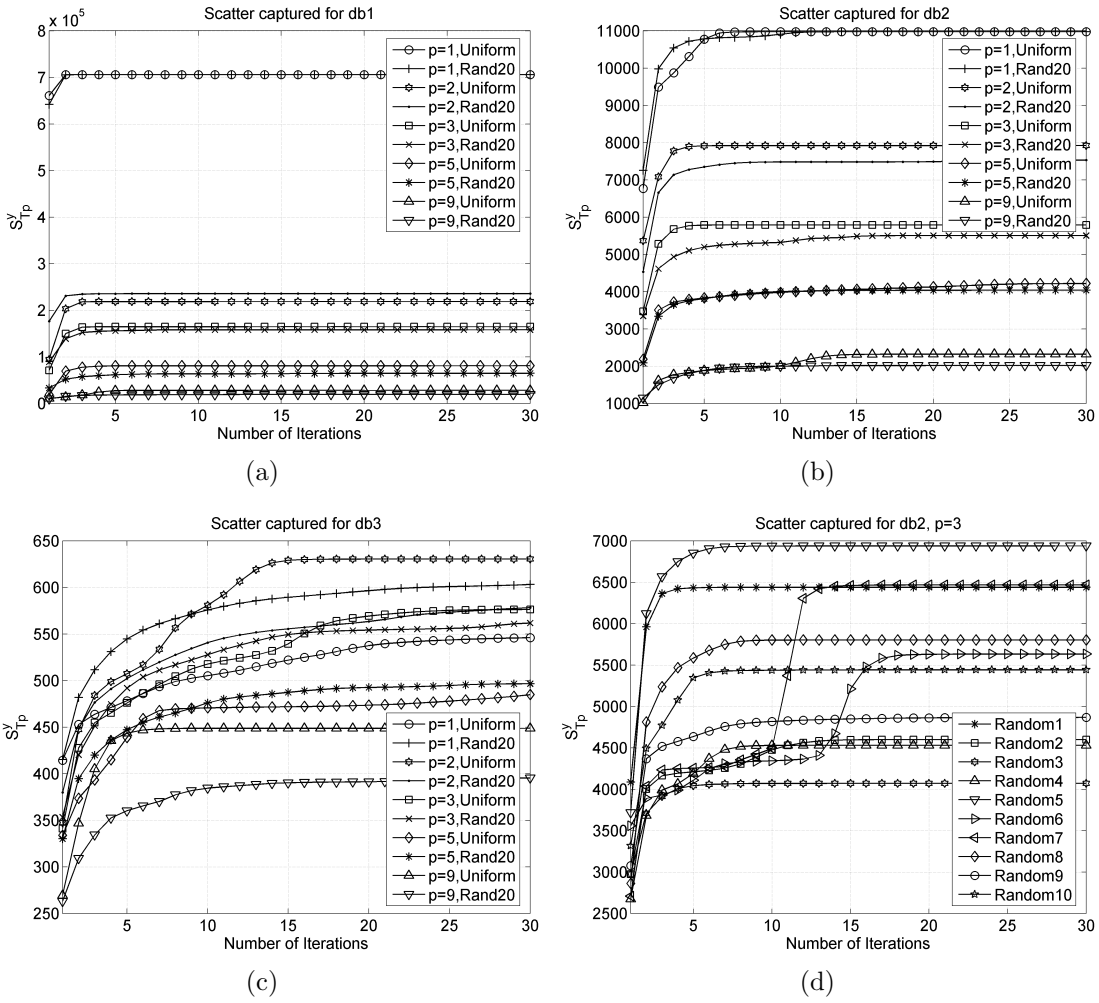


Figure 5.2: Illustration of the effects of initialization on the scatter captured by UMPCA: Comparison of the captured $S^{\mathbf{y}}_{T_p}$ with uniform and random initialization (averaged of 20 repetitions) over 30 iterations for $p = 1, 2, 3, 5, 9$ on synthetic data set (a) db1, (b) db2, and (c) db3; (d) Illustration of the captured $S^{\mathbf{y}}_{T_p}$ of 10 random initializations for $p = 3$ on db2.

5.3.1 The effects of initialization and projection order

The effects of initialization are studied first, with the uniform initialization and random initialization tested up to 30 iterations and the projection order fixed to be [1 2 3]. Figure 5.2 shows the simulation results on the three synthetic data sets. The results shown for

random initialization in Figs. 5.2(a), 5.2(b), and 5.2(c) are the average of 20 repeated trials. From Figs. 5.2(a) and 5.2(b), it can be seen that for $p = 1$, both the uniform and random initializations result in the same $S^{\mathbf{y}}_{T_p}$. For $p > 1$, two ways of initialization lead to different $S^{\mathbf{y}}_{T_p}$, with the uniform initialization performs better (i.e., results in larger $S^{\mathbf{y}}_{T_p}$) on db2. In addition, it should be noted that for db1 and db2, $S^{\mathbf{y}}_{T_p}$ decreases as p increases, and the algorithm converges in around 5 and 15 iterations for db1 and db2, respectively. While for db3 in Fig. 5.2(c), the uniform and random initialization do not results in the same $S^{\mathbf{y}}_{T_p}$ even for $p = 1$ and $S^{\mathbf{y}}_{T_p}$ does not always decrease as p increases. This unusual behavior on db3 matches the difficulty observed in Section 4.6 (page 86) for MPCA. It may be partly explained by observing from Fig. 5.2(c) that the algorithm converge slowly and 30 iterations may not be sufficient to reach convergence. As pointed out in Section 4.6 (page 86), practical data such as face and gait data shares similar characteristics with db1.

Figure 5.2(d) further shows some typical results of the evolution of $S^{\mathbf{y}}_{T_p}$ for ten random initializations on db2 with $p = 3$. As seen from the figure, the results obtained from random initialization have high variance. Therefore, the uniform initialization is preferred and used in all the following experiments for UMPCA. However, if the computational cost is not important, a number of random initializations can be tested to choose the one results in the best performance, i.e., the largest $S^{\mathbf{y}}_{T_p}$.

Next, the effects of the projection order are tested, with representative results shown in Fig. 5.3 for $p = 1, 2$ on the three synthetic data sets. As shown in the figure, the projection order affects UMPCA as well, except for $p = 1$ on db1 and db2. Nonetheless, no one particular projection order consistently outperforms all the others. Thus, in the following experiments, the projection order is fixed to be sequential from 1 to N. As in the initialization, if computational cost is not a concern, all possible projection orders could be tested and the one results in the largest $S^{\mathbf{y}}_{T_p}$ should be taken for each p.

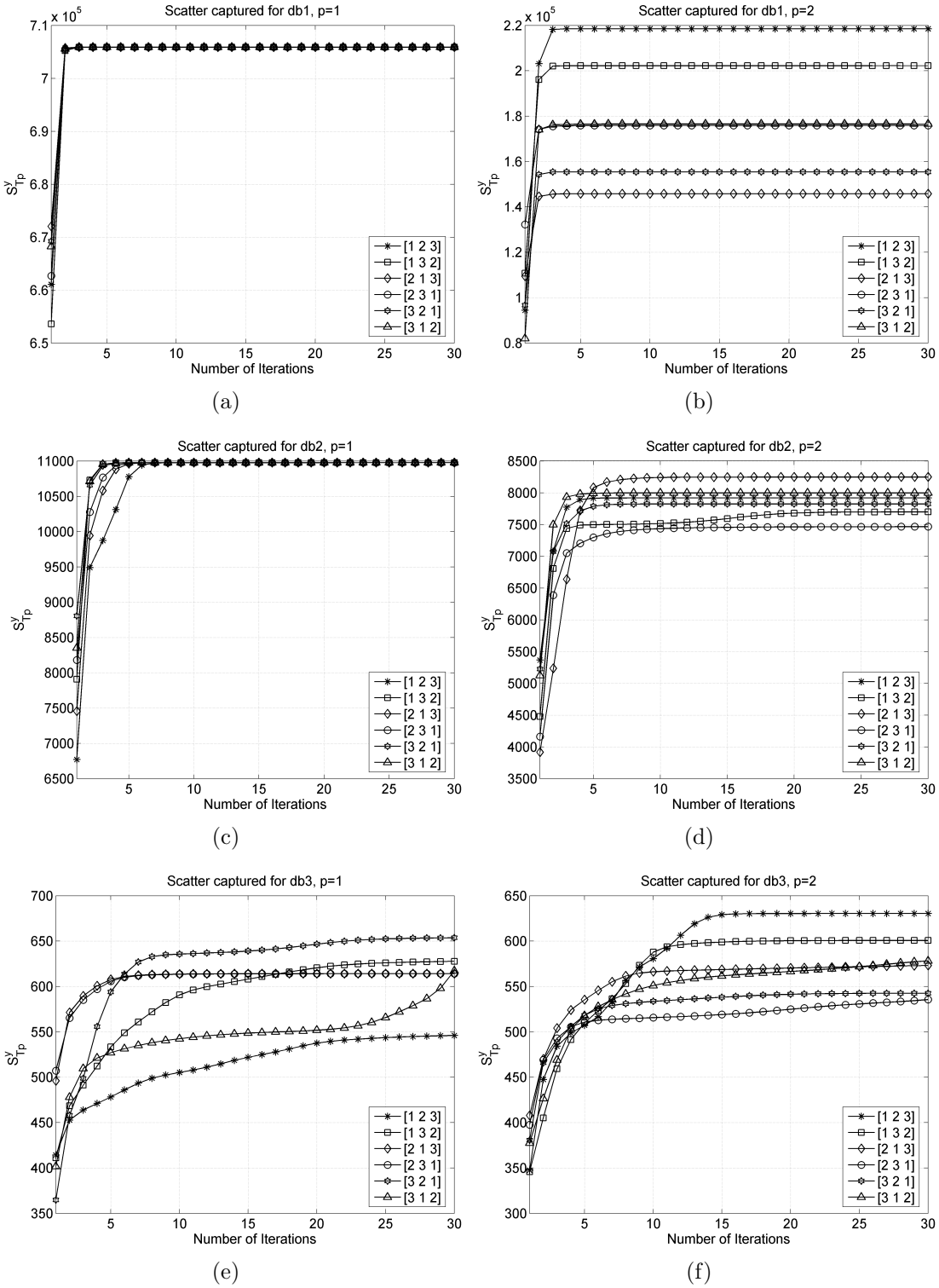


Figure 5.3: Illustration of the effects of projection order on the scatter captured by UMPCA: on db1 with (a) $p = 1$ and (b) $p = 2$, on db2 with (c) $p = 1$ and (d) $p = 2$, on db3 with (e) $p = 1$ and (f) $p = 2$.

5.3.2 Convergence studies

Lastly, this subsection studies the convergence of the total scatter captured in each elementary multilinear projection and the convergence of the corresponding projection vectors in each mode. Figure 5.4 depicts the evolution of the captured total scatter and the 2-mode projection vector difference over 50 iterations for $p = 1, ..., 10$. From Figs. 5.4(a), 5.4(c), and 5.4(e), it can be observed that the algorithm converges (in terms of the total scatter) on db1 and db2 in around 10 and 30 iterations, respectively, while on db3, the convergence speed is considerably lower in general, indicating again the difficulty of db3. Figures 5.4(b), 5.4(d), and 5.4(f) demonstrate that the projection vectors obtained converge as well though in some cases, they may converge slower than the total scatter captured. In addition, it is again observed that the convergence speed of the projection vectors on db3 is also much lower than the other two data sets.

5.4 Summary

In this chapter, the UMPCA algorithm is derived, where uncorrelated features are extracted directly from tensorial data through the tensor-to-vector projection. The algorithm successively maximizes variance captured by each elementary projection while enforcing the zero-correlation constraint. As in MPCA, the UMPCA solution employs alternating projection and is iterative too. The connections to existing solutions are pointed out and design issues such as initialization and termination are discussed, following by the analysis on the computational complexity and memory requirement. Experimental studies on the properties of UMPCA are carried out on the synthetic data sets used in the studies of MPCA and the results give some guidance on the settings of UMPCA in practice.

The MPCA and UMPCA algorithms proposed so far have addressed the two unexplored approaches in Fig. 3.7(a) (page 51). They are both unsupervised learning

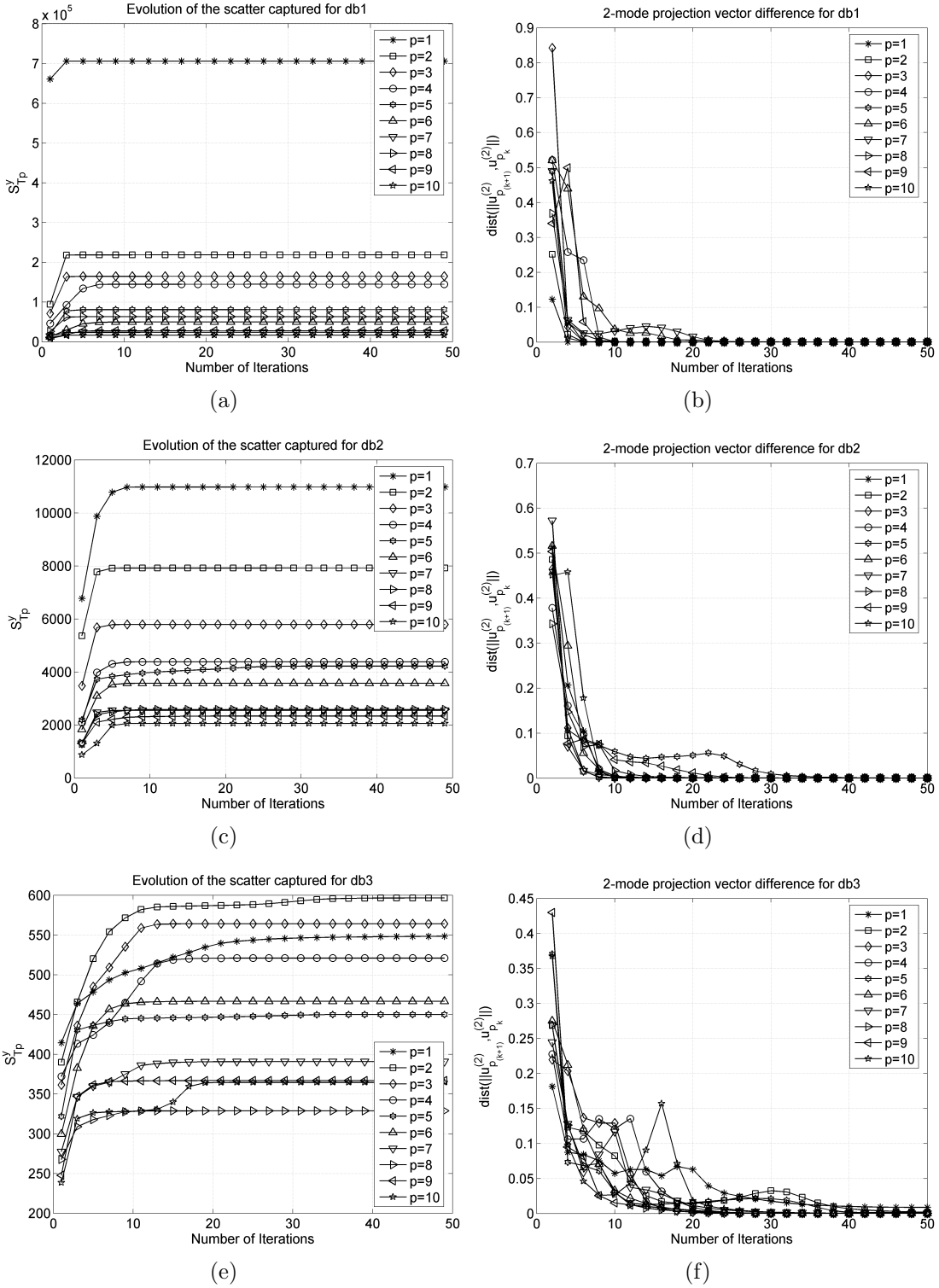


Figure 5.4: Illustration of the convergence of UMPCA: the evolution of the total scatter captured on (a) db1, (c) db2, and (e) db3; and the evolution of the *dist* $\left(\mathbf{u}^{(2)}_{p_{(k)}}, \mathbf{u}^{(2)}_{p_{(k-1)}}\right)$ on (b) db1, (d) db2, and (f) db3.

algorithms, which do not take underlying class structure information into account in the feature extraction process, even when such information is available. They aim to maximize the total scatter, which includes both the within-class scatter and the between-class scatter. While a large between-class scatter is good for separating classes, a large within-class scatter has a negative impact on the classification performance. In the next chapter, the supervised multilinear subspace learning will be explored. Specifically, a new supervised multilinear subspace learning algorithm closely related to UMPCA proposed in this chapter will be developed. It extracts discriminative features by maximizing the between-class scatter while minimizing the within-class scatter, and the zero-correlation constraint is also enforced.

Chapter 6

Uncorrelated Multilinear Discriminant Analysis

6.1 Introduction

This chapter proposes the UMLDA algorithm and its enhancements. On one hand, the focus here is on supervised multilinear subspace learning, which is different from the previous two chapters of unsupervised methods. On the other hand, UMLDA shares similarity with UMPCA in Chapter 5 in the motivation of deriving uncorrelated features that are often desirable in recognition tasks since they contain minimum redundancy and ensure linear independence of features. As in UMPCA, UMLDA aims to achieve the objective through the tensor-to-vector projection rather than the tensor-to-tensor projection. From the discussions in Section 3.4.4 (page 57), the existing supervised multilinear subspace learning algorithms concentrate on discrimination criterion construction while none of them produces uncorrelated features as the classical LDA does. Moreover, as pointed out in Section 3.5 (page 60), in supervised multilinear subspace learning through the tensor-to-vector projection, the approach of scatter ratio maximization has not been studied. The only method in this category tries to maximize the scatter difference through

a heuristic approach as in the unsupervised method TROD.

Therefore, the proposed UMLDA aims to extract uncorrelated discriminative features directly from tensorial data through solving a tensor-to-vector projection that maximizes the traditional scatter-ratio-based criterion. It addresses the shaded empty box in Fig. 3.7(b) (page 51) under the tensor-to-vector projection. The solution consists of sequential iterative processes based on the alternating projection method as in UMPCA. In addition, an adaptive regularization factor is incorporated, where the within-class scatter estimation is increased through a data-independent regularization parameter. The regularization targets to enhance the performance in the small sample size scenario, which often causes difficulties in practical recognition applications, especially for supervised algorithms. Furthermore, as different initialization or regularization of UMLDA results in different features, an aggregation scheme is adopted to combine several differently initialized and differently regularized UMLDA feature extractors at the matching score level using the simple sum rule. With the aggregation, the complementary information from differently initialized and regularized UMLDA recognizers are exploited, resulting in enhanced recognition performance while alleviating the regularization parameter selection problem faced in most regularization methods.

The rest of this chapter is organized as follows. In Section 6.2, the UMLDA problem is formulated and an iterative solution is derived, with an adaptive regularization procedure introduced for better generalization in the small sample size scenario. Next, the connections to the existing supervised subspace learning algorithms are discussed. Issues regarding the initialization method, projection order, termination condition, and convergence are also addressed in this section. Section 6.3 presents the matching score level aggregation of multiple UMLDA feature extractors that are differently initialized and regularized to enhance the recognition performance. In Section 6.4, the properties of the proposed UMLDA solutions are illustrated through a set of face recognition experiments. Finally, Section 6.5 summarizes this chapter.

6.2 The UMLDA with Regularization

In this section, UMLDA is formulated and a regularized solution is derived. Its connections with existing solutions, as well as the design and computational issues, are then addressed. Similar to UMPCA, in the presentation, for the convenience of discussion, the training samples are assumed to be zero-mean so that the constraint of uncorrelated features is the same as orthogonal features, from Proposition 5.1 (page 97).

6.2.1 The UMLDA problem

The classical Fisher's discrimination criterion in LDA [3] is defined as the scatter ratio for vector samples and this ratio is extended to scalar samples with Definition 2.5 (page 32). As in UMPCA, the pth projected (scalar) features are $\{y_{m_p}, m = 1, ..., M\}$, where M is the number of training samples and y_{m_p} is the projection of the mth sample $\mathcal{X}_m$ by the pth elementary multilinear projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$:

$$y_{m_p} = \mathcal{X}_m \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}. \tag{6.1}$$

Their corresponding between-class scatter $S_{B_p}^{\mathbf{y}}$ and the within-class scatter $S_{W_p}^{\mathbf{y}}$ are then

$$S_{B_p}^{\mathbf{y}} = \sum_{c=1}^{C} M_c(\bar{y}_{c_p} - \bar{y}_p)^2, \;\; S_{W_p}^{\mathbf{y}} = \sum_{m=1}^{M} (y_{m_p} - \bar{y}_{c_{m_p}})^2, \tag{6.2}$$

where C is the number of classes, M_c is the number of samples for class c, c_m is the class label for the mth training sample, $\bar{y}_p = \frac{1}{M}\sum_m y_{m_p} = 0$ and $\bar{y}_{c_p} = \frac{1}{M_c}\sum_{m,c_m=c} y_{m_p}$. Thus, the Fisher's discrimination criterion for the pth scalar samples is

$$F_p^{\mathbf{y}} = \frac{S_{B_p}^{\mathbf{y}}}{S_{W_p}^{\mathbf{y}}}. \tag{6.3}$$

Also, let $\mathbf{g}_p$ denote the pth coordinate vector. Its mth component $\mathbf{g}_p(m) = y_{m_p}$.

A formal definition of the multilinear subspace learning problem to be solved in UMLDA is then given in the following.

A set of M tensor object samples $\{\mathcal{X}_1,\ \mathcal{X}_2,\ ...,\ \mathcal{X}_M\}$ (with zero-mean) is available for training. Each tensor object $\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ assumes values in the tensor space $\mathbb{R}^{I_1} \bigotimes \mathbb{R}^{I_2} ... \bigotimes \mathbb{R}^{I_N}$. The objective of UMLDA is to find a tensor-to-vector projection, which consists of P elementary multilinear projections $\{\mathbf{u}_p^{(n)} \in \mathbb{R}^{I_n \times 1}, n = 1, ..., N\}_{p=1}^{P}$, 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 $P < \prod_{n=1}^{N} I_n$):

$$\mathbf{y}_m = \mathcal{X}_m \times_{n=1}^{N} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}, m = 1, ..., M, \tag{6.4}$$

such that the Fisher's discrimination criterion $F_p^{\mathbf{y}}$ is maximized in each elementary multilinear projection direction, subject to the constraint that the P coordinate vectors $\{\mathbf{g}_p \in \mathbb{R}^M, p = 1, ..., P\}$ are uncorrelated.

In other words, the UMLDA objective is to determine a set of P elementary multilinear projections $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ that maximize the scatter ratio while producing features with zero-correlation. Thus, the objective function for the pth elementary multilinear projection is

$$\begin{aligned} \{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\} &= \arg\max F_p^{\mathbf{y}}, \\ \texttt{subject to } \frac{\mathbf{g}_p^T \mathbf{g}_q}{\| \mathbf{g}_p \| \| \mathbf{g}_q \|} &= \delta_{pq},\ p, q = 1, ..., P, \end{aligned} \tag{6.5}$$

where δ_{pq} is the Kronecker delta defined in (5.4) (page 98).

6.2.2 The derivation of Regularized UMLDA (R-UMLDA)

To solve the problem, the successive determination approach in the derivation of the ULDA in [45] is followed, similar to the successive approach in UMPCA. The P ele-

mentary multilinear projections $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^{P}$ are determined sequentially in P steps, with the pth step obtaining the pth elementary multilinear projection:

Step 1: Determine the first elementary multilinear projection $\{\mathbf{u}_1^{(n)^T}, n = 1, ..., N\}$ by maximizing $F_1^{\mathbf{y}}$ without any constraint.

Step 2: Determine the second elementary multilinear projection $\{\mathbf{u}_2^{(n)^T}, n = 1, ..., N\}$ by maximizing $F_2^{\mathbf{y}}$ subject to the constraint that $\mathbf{g}_2^T\mathbf{g}_1 = 0$.

Step 3: Determine the third elementary multilinear projection $\{\mathbf{u}_3^{(n)^T}, n = 1, ..., N\}$ by maximizing $F_3^{\mathbf{y}}$ subject to the constraint that $\mathbf{g}_3^T\mathbf{g}_1 = 0$ and $\mathbf{g}_3^T\mathbf{g}_2 = 0$.

Step p ($p = 4, ..., P$): Determine the pth elementary multilinear projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$ by maximizing $F_p^{\mathbf{y}}$ subject to the constraint that $\mathbf{g}_p^T\mathbf{g}_q = 0$ for $q = 1, ..., p-1$.

The algorithm to compute these elementary multilinear projections is summarized in the pseudo-code in Fig. 6.1. The detailed derivation is presented below.

As in UMPCA, to solve for the pth elementary multilinear projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}$, N sets of parameters corresponding to N projection vectors, $\mathbf{u}_p^{(1)}, \mathbf{u}_p^{(2)}, ...\mathbf{u}_p^{(N)}$, need to be determined, one in each mode. However, simultaneous determination of these N sets of parameters in all modes is a complicated nonlinear problem without an existing optimal solution, except when $N = 1$, the classical linear case where only one projection vector is to be solved.

Therefore, this typical multilinear problem is solved again by the alternating projection method. For each elementary multilinear projection to be determined, the parameters of the projection vector $\mathbf{u}_p^{(n^*)}$ for each mode n^* are estimated one mode by one mode separately, conditioned on $\{\mathbf{u}_p^{(n)}, n \neq n^*\}$, the parameter values of the projection vectors in the other modes. Each conditional subproblem is linear and depends only on $\mathbf{u}_p^{(n^*)}$. This iterative process corresponds to the loop indexed by k in Fig. 6.1. In each iteration k, the loop indexed by n in Fig. 6.1 consists of the N conditional subproblems.

Input: A set of zero-mean tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}, m = 1, ..., M\}$ with class labels $\mathbf{c} \in \mathbb{R}^M$, the desired feature vector length P, the regularization parameter γ, the maximum number of iterations K, and a small number ϵ for testing convergence.

Output: The P elementary multilinear projections $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1}^P$ that best separate classes in the projected space.

Algorithm:

For $p = 1 : P$ (step p: determine the pth elementary multilinear projections)

If $p > 1$, calculate the coordinate vector $\mathbf{g}_{p-1}$: $\mathbf{g}_{p-1}(m) = \mathcal{X}_m \times_1 \mathbf{u}_{p-1}^{(1)^T} \times_2 \mathbf{u}_{p-1}^{(2)^T} ... \times_N \mathbf{u}_{p-1}^{(N)^T}$.

- For $n = 1, ..., N$, initialize $\mathbf{u}_{p_{(0)}}^{(n)} \in \mathbb{R}^{I_n}$.
- For $k = 1 : K$
  - For $n = 1 : N$
    - Calculate $\tilde{\mathbf{y}}_{m_p}^{(n)} = \mathcal{X}_m \times_1 \mathbf{u}_{p_{(k)}}^{(1)^T} ... \times_{n-1} \mathbf{u}_{p_{(k)}}^{(n-1)^T} \times_{n+1} \mathbf{u}_{p_{(k-1)}}^{(n+1)^T} ... \times_N \mathbf{u}_{p_{(k-1)}}^{(N)^T}$, for $m = 1, ..., M$.
    - Calculate $\mathbf{R}_p^{(n)}$, $\tilde{\mathbf{S}}_{B_p}^{(n)}$, and $\tilde{\mathbf{S}}_{W_p}^{(n)}$. Set $\mathbf{u}_{p_{(k)}}^{(n)}$ to be the (unit) eigenvector of $\left(\tilde{\mathbf{S}}_{W_p}^{(n)}\right)^{-1} \mathbf{R}_p^{(n)} \tilde{\mathbf{S}}_{B_p}^{(n)}$ associated with the largest eigenvalue.
  - If $k = K$ or $dist\left(\mathbf{u}_{p_{(k)}}^{(n)}, \mathbf{u}_{p_{(k-1)}}^{(n)}\right) < \epsilon$ for all n, set $\mathbf{u}_p^{(n)} = \mathbf{u}_{p_k}^{(n)}$ for all n, break.
- **Output** $\{\mathbf{u}_p^{(n)}\}$. Go the step $p + 1$ if $p < P$. Stop if $p = P$.

Figure 6.1: The pseudo-code implementation of the R-UMLDA algorithm for feature extraction from tensor objects.

To solve for $\mathbf{u}_p^{(n^*)}$ in the n^*-mode, assuming that $\{\mathbf{u}_p^{(n)}, n \neq n^*\}$ is given, the tensor samples are projected in these $(N-1)$ modes $\{n \neq n^*\}$ first to obtain

$$\tilde{\mathbf{y}}_{m_p}^{(n^*)} = \mathcal{X}_m \times_1 \mathbf{u}_p^{(1)^T} ... \times_{n^*-1} \mathbf{u}_p^{(n^*-1)^T} \times_{n^*+1} \mathbf{u}_p^{(n^*+1)^T} ... \times_N \mathbf{u}_p^{(N)^T}, \tag{6.6}$$

$\tilde{\mathbf{y}}_{m_p}^{(n^*)} \in \mathbb{R}^{I_{n^*}}$. The conditional subproblem then becomes to determine $\mathbf{u}_p^{(n^*)}$ that projects the vector samples $\{\tilde{\mathbf{y}}_{m_p}^{(n^*)}, m = 1, ..., M\}$ onto a line so that the scatter ratio is maximized, subject to the zero-correlation constraint. This is a (linear and simpler) ULDA problem with the input samples $\{\tilde{\mathbf{y}}_{m_p}^{(n^*)}, m = 1, ..., M\}$. The corresponding between-class scatter matrix $\tilde{\mathbf{S}}_{B_p}^{(n^*)}$ and the (regularized) within-class scatter matrix $\tilde{\mathbf{S}}_{W_p}^{(n^*)}$ are then defined as

$$\tilde{\mathbf{S}}_{B_p}^{(n^*)} = \sum_{c=1}^{C} M_c (\bar{\tilde{\mathbf{y}}}_{c_p}^{(n^*)} - \bar{\tilde{\mathbf{y}}}_p^{(n^*)})(\bar{\tilde{\mathbf{y}}}_{c_p}^{(n^*)} - \bar{\tilde{\mathbf{y}}}_p^{(n^*)})^T, \tag{6.7}$$

$$\tilde{\mathbf{S}}_{W_p}^{(n^*)} = \sum_{m=1}^{M} (\tilde{\mathbf{y}}_{m_p}^{(n^*)} - \bar{\tilde{\mathbf{y}}}_{c_{m_p}}^{(n^*)})(\tilde{\mathbf{y}}_{m_1}^{(n^*)} - \bar{\tilde{\mathbf{y}}}_{c_{m_p}}^{(n^*)})^T + \gamma \cdot \lambda_{max}(\breve{\mathbf{S}}_W^{(n^*)}) \cdot \mathbf{I}_{I_{n^*}}, \tag{6.8}$$

where $\bar{\tilde{\mathbf{y}}}_{c_p}^{(n^*)} = \frac{1}{M_c} \sum_{m, c_m = c} \tilde{\mathbf{y}}_{m_p}^{(n^*)}$, $\bar{\tilde{\mathbf{y}}}_p^{(n^*)} = \frac{1}{M} \sum_m \tilde{\mathbf{y}}_{m_p}^{(n^*)} = \mathbf{0}$, $\gamma \geq 0$ is a regularization parameter, $\mathbf{I}_{I_{n^*}}$ is an identity matrix of size $I_{n^*} \times I_{n^*}$, and $\lambda_{max}(\breve{\mathbf{S}}_W^{(n^*)})$ is the maximum eigenvalue of $\breve{\mathbf{S}}_W^{(n^*)}$, the within-class scatter matrix for the n-mode vectors of the training samples, defined as

$$\breve{\mathbf{S}}_W^{(n^*)} = \sum_{m=1}^{M} \left(\mathbf{X}_{m(n^*)} - \bar{\mathbf{X}}_{c_m(n^*)}\right) \left(\mathbf{X}_{m(n^*)} - \bar{\mathbf{X}}_{c_m(n^*)}\right)^T, \tag{6.9}$$

where $\bar{\mathbf{X}}_{c(n^*)}$ is the n^*-mode unfolded matrix of the class mean $\bar{\mathcal{X}}_c = \frac{1}{M_c} \sum_{m, c_m = c} \mathcal{X}_m$. In the following, the motivation for introducing the regularization factor is explained.

In the targeted applications of face and gait recognition (and many other applications as well), the dimensionality of the input data is very high while at the same time, the number of training samples for each class is often too small to represent the true characteristics of their classes, resulting in the well-known small sample size problem [92].

Furthermore, empirical study of the iterative UMLDA algorithm (i.e., $\gamma = 0$) under the small sample size scenario indicates that the iterations tend to minimize the within-class scatter towards zero in order to maximize the scatter ratio since the scatter ratio reaches maximum of infinity when the within-class scatter is zero and the between-class scatter is non-zero. However, the estimated within-class scatter on the training data is usually much smaller than the real within-class scatter, due to limited number of samples for each class. Therefore, regularization [29], which has been routinely used for combatting the singularity problem in the LDA-based algorithms under the small sample size scenario [69, 92], is adopted here to improve the generalization capability of UMLDA under the small sample size scenario, leading to the R-UMLDA algorithm. The regularization term is introduced in (6.8) so that during the iteration, less focus is put on shrinking the within-class scatter. Moreover, the regularization introduced is adaptive since γ is the only regularization parameter and the regularization term in the n^*-mode is scaled by $\lambda_{max}(\breve{\mathbf{S}}_W^{(n^*)})$, an approximate estimate of the n^*-mode within-class scatter in the training data. The basic UMLDA is obtained by setting $\gamma = 0$.

With (6.7) and (6.8), it is ready to solve the P elementary multilinear projections. For $p = 1$, the $\mathbf{u}_1^{(n^*)}$ that maximizes the Fisher's discrimination criterion

$$\frac{\mathbf{u}_1^{(n^*)^T} \tilde{\mathbf{S}}_{B_1}^{(n^*)} \mathbf{u}_1^{(n^*)}}{\mathbf{u}_1^{(n^*)^T} \tilde{\mathbf{S}}_{W_1}^{(n^*)} \mathbf{u}_1^{(n^*)}} \tag{6.10}$$

in the projected space is obtained as the unit eigenvector of

$$\left(\tilde{\mathbf{S}}_{W_1}^{(n^*)}\right)^{-1} \tilde{\mathbf{S}}_{B_1}^{(n^*)} \tag{6.11}$$

associated with the largest eigenvalue for a nonsingular $\tilde{\mathbf{S}}_{W_1}^{(n^*)}$. Next, given the first $(p-1)$ elementary multilinear projections, where $p > 1$, the pth elementary multilinear projection aims to maximize the scatter ratio $F_p^{\mathbf{y}}$, subject to the constraint that features projected by the pth elementary multilinear projection are uncorrelated with those pro-

jected by the first $(p-1)$ elementary multilinear projections. Again, let $\tilde{\mathbf{Y}}_p^{(n^*)} \in \mathbb{R}^{I_{n^*} \times M}$ be a matrix with its mth column to be $\tilde{\mathbf{y}}_{m_p}^{(n^*)}$, i.e., $\tilde{\mathbf{Y}}_p^{(n^*)} = \left[\tilde{\mathbf{y}}_{1_p}^{(n^*)}, \tilde{\mathbf{y}}_{2_p}^{(n^*)}, ..., \tilde{\mathbf{y}}_{M_p}^{(n^*)}\right]$. The pth coordinate vector is then obtained as $\mathbf{g}_p = \tilde{\mathbf{Y}}_p^{(n^*)^T} \mathbf{u}_p^{(n^*)}$. The constraint that $\mathbf{g}_p$ is uncorrelated with $\{\mathbf{g}_q, q = 1, ..., p-1\}$ can be written as

$$\mathbf{g}_p^T \mathbf{g}_q = \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = 0, q = 1, ..., p-1. \tag{6.12}$$

Thus, $\mathbf{u}_p^{(n^*)}$ $(p > 1)$ can be determined by solving the following constrained optimization problem:

$$\begin{aligned} \mathbf{u}_p^{(n^*)} &= \arg\max \frac{\mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)}}{\mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)} \mathbf{u}_p^{(n^*)}}, \\ \texttt{subject to} \quad & \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = 0, q = 1, ..., p-1. \end{aligned} \tag{6.13}$$

The solution is given by the following theorem for nonsingular $\tilde{\mathbf{S}}_{W_p}^{(n^*)}$:

Theorem 6.1. *When $\tilde{\mathbf{S}}_{W_p}^{(n^*)}$ is nonsingular, the solution to the problem (6.13) is the (unit-length) generalized eigenvector corresponding to the largest generalized eigenvalue of the following generalized eigenvalue problem:*

$$\mathbf{R}_p^{(n^*)} \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u} = \lambda \tilde{\mathbf{S}}_{W_p}^{(n^*)} \mathbf{u}, \tag{6.14}$$

where

$$\mathbf{R}_p^{(n^*)} = \mathbf{I}_{I_{n^*}} - \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \left(\mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \right)^{-1} \mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}, \tag{6.15}$$

$$\mathbf{G}_{p-1} = [\mathbf{g}_1 \quad \mathbf{g}_2 \quad ... \mathbf{g}_{p-1}] \in \mathbb{R}^{M \times (p-1)}. \tag{6.16}$$

Proof. The proof of Theorem 6.1 is given in Appendix A.7. □

By setting $\mathbf{R}_1^{(n^*)} = \mathbf{I}_{I_{n^*}}$ and from Theorem 6.1, a unified solution for R-UMLDA is obtained when $\tilde{\mathbf{S}}_{W_p}^{(n^*)}$ is nonsingular: for $p = 1, ..., P$, $\mathbf{u}_p^{(n^*)}$ is obtained as the unit eigenvector of

$$\left(\tilde{\mathbf{S}}_{W_p}^{(n^*)}\right)^{-1} \mathbf{R}_p^{(n^*)} \tilde{\mathbf{S}}_{B_p}^{(n^*)} \tag{6.17}$$

associated with the largest eigenvalue. In addition, as in UMPCA, the maximum number of features that can be extracted by R-UMLDA does not exceed $\min\{\min_n I_n, M\}$, similarly from Corollary 5.1 (page 103).

6.2.3 Connections with existing solutions

This subsection discusses the connections of UMLDA (i.e., no regularization) with existing supervised subspace learning algorithms.

The UMLDA algorithm follows the approach of successive scatter ratio maximization of the ULDA derivation in [45], proven to be equivalent to the classical LDA [46, 158]. Hence, when $N = 1$, UMLDA reduces to LDA, which can also been seen from Section 2.2.4 (page 25).

From Section 3.4.4 (page 57), TR1DA [142, 123] [116] shares similarity with UMLDA in that it also seeks for a tensor-to-vector projection to maximize some scatter measure (difference), with a heuristic greedy approach of residue calculation as in TROD [116]. In contrast, UMLDA takes a systematic, more principled formulation with the correlation among features taken into consideration.

6.2.4 Initialization, projection order, termination, and convergence

This subsection discusses the various implementation issues of R-UMLDA, in the order of the algorithm flow in Fig. 6.1: initialization, projection order, termination, and convergence.

As in UMPCA, the R-UMLDA algorithm is iterative, and initial estimations for the projection vectors $\{\mathbf{u}_p^{(n)}\}$ are necessary. However, the determination of the optimal initialization in R-UMLDA is still an open problem too, with no guidance from either the algorithm or the data on the best initialization that could result in the best separation of the classes in the feature space. The uniform initialization and random initialization mentioned in Section 5.2.4 (page 104) are empirically studied here as well. The empirical studies in Section 6.4 indicate that the results of R-UMLDA are affected by initialization, and the uniform initialization gives better results.

The mode ordering in computing the projection vectors also affects the solution. Similar to initialization, there is no way to determine the optimal projection order either and it is considered to be an open problem as well. Empirical studies on the effects of the projection order indicate that with all the other algorithm settings fixed, altering the projection order does result in some performance differences, but there is no guidance from either the data or the algorithm on what projection order in the iteration leads to the best separation of classes. Therefore, there is no preference on a particular projection order and in practice, the projection vectors are solved sequentially (from 1-mode to N-mode), as in UMPCA.

Remark 6.1. *Although the optimal initialization and the optimal projection order cannot be determined, the aggregation scheme suggested in Section 6.3 reduces the significance of their optimal determination.*

As seen from Fig. 6.1, the termination criterion can be simply set to a maximum number of iterations K or it can be set by examining the convergence of the projection vectors:

$$dist\left(\mathbf{u}_{p_{(k)}}^{(n)}, \mathbf{u}_{p_{(k-1)}}^{(n)}\right) < \epsilon, \tag{6.18}$$

where ϵ is a user-defined small number threshold, and this distance is defined as in (5.16) (page 105). As to be shown in Sec. 6.4, the recognition performance increases slowly

after the first a few iterations. Therefore, the iteration can be terminated by setting K in practice for convenience, especially when the computational cost is a concern.

Regarding the convergence, the derivation of Theorem 6.1 (Appendix A.7) implies that per iteration, the scatter ratio $F_p^{\mathbf{y}}$ is a non-decreasing function (as it either remains the same or increases) since each update of the projection vector $\mathbf{u}^{(n^*)}$ in a given mode n^* maximizes $F_p^{\mathbf{y}}$, while the projection vectors in all the other modes, $\{\mathbf{u}^{(n)}, n \neq n^*\}$, are considered fixed. However, the ratio $F_p^{\mathbf{y}}$ may not have an upper-bound as in MPCA and UMPCA since it may reach infinity if there exists a projection that can lead to zero within-class scatter, especially when there are only a small number of samples and the regularization is not strong enough. Therefore, R-UMLDA may not converge in terms of $F_p^{\mathbf{y}}$ in all cases. Simulation studies presented in Section 6.4 indicate that in practice, the projection vector $\{\mathbf{u}^{(n)}\}$ obtained by the R-UMLDA algorithm converges within 10 iterations for facial objects with strong regularization, where the convergence is up to sign.

6.2.5 Computational aspects of UMLDA

Next, the computational complexity and memory requirements of UMLDA are analyzed. It is assumed again that $I_1 = I_2 = ... = I_N = \left(\prod_{n=1}^{N} I_n\right)^{\frac{1}{n}} = I$ for simplicity in the analysis.

For the computational complexity, the most computational demanding steps are the calculations of the projection $\tilde{\mathbf{y}}_{m_p}^{(n)}$, the computation of $\tilde{\mathbf{S}}_{B_p}^{(n)}$, $\tilde{\mathbf{S}}_{W_p}^{(n)}$, and $\mathbf{R}_p^{(n)}$, and the calculation of the leading eigenvector of $\left(\tilde{\mathbf{S}}_{W_p}^{(n)}\right)^{-1} \mathbf{R}_p^{(n)} \tilde{\mathbf{S}}_{B_p}^{(n)}$. The complexity of calculating $\tilde{\mathbf{y}}_{m_p}^{(n)}$ for $m = 1, ..., M$, $\tilde{\mathbf{S}}_{B_p}^{(n)}$, and $\tilde{\mathbf{S}}_{W_p}^{(n)}$ are in order of $O(M \cdot \sum_{n=2}^{N} I^n)$, $O(C \cdot I^2)$, and $O(M \cdot I^2)$, respectively. The computation of $\mathbf{R}_p^{(n)}$ is in order of

$$\begin{aligned} & O\left(I \cdot M \cdot (p-1) + I^3 + 2 \cdot (p-1) \cdot I^2 + (p-1)^3 + 2 \cdot I \cdot (p-1)^2\right) \\ = \; & O\left(I^3 + (p-1) \cdot \left[I \cdot M + 2 \cdot I^2 + (p-1)^2 + 2 \cdot I \cdot (p-1)\right]\right). \end{aligned} \tag{6.19}$$

Lastly, the computation of $\left(\tilde{\mathbf{S}}_{W_p}^{(n)}\right)^{-1}\mathbf{R}_p^{(n)}\tilde{\mathbf{S}}_{B_p}^{(n)}$ and its eigen-decomposition[1] are of order $O(2\cdot I^3)$ and $O(I^3)$, respectively. Therefore, the computational complexity per mode n for one iteration k of step p is

$$O\left(M\sum_{n=2}^{N}I^n+(C+M)I^2+(p-1)\left[I\cdot M+2I^2+(p-1)^2+2I(p-1)\right]+4I^3\right). \tag{6.20}$$

For the memory requirement, as in UMPCA and MPCA, the respective computation can be done incrementally by reading $\mathcal{X}_m$ sequentially. Hence, the memory needed for the UMLDA algorithm can be as low as $O(I^N)$ except for $N=1$ although sequential reading will lead to higher I/O cost.

From the discussions above, similar to UMPCA, the UMLDA algorithm obtains the solution through a sequential iterative procedure and this may lead to a high computational and I/O cost. However, this is not considered a disadvantage since solving the UMLDA projection is only in the training phase and it can be done offline. In the testing phase, the extraction (projection) of features from a test sample is also an efficient linear operation.

6.3 Aggregation of R-UMLDA Recognizers

This section proposes the aggregation of a number of differently initialized and regularized UMLDA recognizers for enhanced performance, motivated from two properties of the basic recognizer using R-UMLDA as the feature extractor. On one hand, by Corollary 5.1 (page 103) and as to be shown in Sec. 6.4, the number of useful discriminative features that can be extracted by a single R-UMLDA is limited. On the other hand, since R-UMLDA is affected by initialization and regularization, which cannot be optimally determined, different initialization or regularization could result in different discriminative

[1]UMLDA also needs only the largest eigenvalue and the corresponding eigenvector, so more efficient computational methods may be applied in practice.

features (also see Sec. 6.4). From the generalization theory explaining the success of random subspace method [39], bagging, and boosting [111, 120, 9], the sensitivity of R-UMLDA to initialization and regularization suggests that R-UMLDA is not a very stable learner (feature extractor) and it is good for ensemble-based learning. Therefore, the aggregation of several differently initialized and regularized UMLDA feature extractors is proposed to get the regularized UMLDA with aggregation (R-UMLDA-A) recognition system so that multiple R-UMLDA recognizers can work together to achieve better recognition performance.

Remark 6.2. *Different projection orders could also result in different features so R-UMLDA with different projection orders could be aggregated as well. However, since the effects of projection orders are similar to those of initializations and the number of possible projection orders ($N!$) is much less than the number of possible initializations (infinite), the projection order is fixed and only the initialization and regularization procedures are varied in this dissertation.*

There are various ways to combine (or fuse) several extracted features, including the feature level fusion [104], fusion at the matching score level [105, 52], and more advanced ensemble-based learning such as boosting [111,93,80]. In the R-UMLDA-A proposed here, the simple sum rule in combining matching scores is used although more sophisticated method such as boosting is expected to achieve better results.

Since high diversity of the learners to be combined is preferred in ensemble-based learning [93], both uniform and random initializations are used in R-UMLDA-A for more diversity. In this way, although the best initialization cannot be determined, several R-UMLDAs with different initializations are aggregated to make complementary discriminative features working together to separate classes better. Furthermore, to introduce even more diversity and alleviate the problem of regularization parameter selection at the same time, this work proposes to sample the regularization parameter γ_a from an interval $[10^{-7}, 10^{-2}]$, empirically chosen to cover a wide range of γ, uniformly in log scale

so that each feature extractor is differently regularized, where $a = 1, ..., A$ is the index of the individual R-UMLDA feature extractor and A is the number of R-UMLDA feature extractors to be aggregated.

Figure 6.2 provides the pseudo-code implementation for R-UMLDA-A for tensor object recognition. The input training samples $\{\mathcal{X}_m\}$ are fed into A differently initialized and regularized UMLDA feature extractors described in Fig. 6.1 with parameters P, K, and γ_a to obtain a set of A tensor-to-vector projections

$$\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1_{(a)}}^{P}, a = 1, ..., A. \tag{6.21}$$

The training samples $\{\mathcal{X}_m\}$ are then projected to R-UMLDA feature vectors $\{\mathbf{y}_{m_{(a)}}\}$ using the obtained tensor-to-vector projections. To classify a test sample $\mathcal{X}$, it is projected to A feature vectors $\{\mathbf{y}_{(a)}\}$ using the A tensor-to-vector projections first. Next, for the ath R-UMLDA feature extractor, the nearest-neighbor distance of the test sample $\mathcal{X}$ to each candidate class c is calculated as:

$$d(\mathcal{X}, c, a) = \min_{m, c_m = c} \parallel \mathbf{y}_{(a)} - \mathbf{y}_{m_{(a)}} \parallel . \tag{6.22}$$

The range of $d(\mathcal{X}, c, a)$ is then matched to the interval $[0, 1]$ as:

$$\tilde{d}(\mathcal{X}, c, a) = \frac{d(\mathcal{X}, c, a) - \min_c d(\mathcal{X}, c, a)}{\max_c d(\mathcal{X}, c, a) - \min_c d(\mathcal{X}, c, a)}. \tag{6.23}$$

Finally, the aggregated nearest-neighbor distance is obtained employing the simple sum rule as:

$$d(\mathcal{X}, c) = \sum_{a=1}^{A} \tilde{d}(\mathcal{X}, c, a), \tag{6.24}$$

and the test sample $\mathcal{X}$ is assigned the label:

$$c^* = \arg\min_c d(\mathcal{X}, c). \tag{6.25}$$

Input: A set of zero-mean tensor samples $\{\mathcal{X}_m \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}, m = 1, ..., M\}$ with class labels $\mathbf{c} \in \mathbb{R}^M$, a test tensor sample $\mathcal{X}$, the desired feature vector length P, the R-UMLDA feature extractor (Fig. 6.1), the maximum number of iterations K, the number of R-UMLDA to be aggregated A.

Output: The class label for $\mathcal{X}$.

R-UMLDA-A algorithm:

Step 1. Feature extraction

- For $a = 1 : A$
  - Obtain the ath tensor-to-vector projection $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1_{(a)}}^{P}$ from the ath R-UMLDA (Fig. 6.1) with the input: $\{\mathcal{X}_m\}$, P, K, γ_a, using random or uniform initialization.
  - Project $\{\mathcal{X}_m\}$ and $\mathcal{X}$ to $\{\mathbf{y}_{m_{(a)}}\}$ and $\mathbf{y}_{(a)}$, respectively, using $\{\mathbf{u}_p^{(n)^T}, n = 1, ..., N\}_{p=1_{(a)}}^{P}$.

Step 2. Aggregation at the matching score level for classification

- For $a = 1 : A$
  - For $c = 1 : C$
    - Obtain the nearest-neighbor distance $d(\mathcal{X}, c, a)$.
  - Normalize $d(\mathcal{X}, c, a)$ to $[0, 1]$ to get $\tilde{d}(\mathcal{X}, c, a)$.
- Obtain the aggregated distance $d(\mathcal{X}, c)$.
- **Output** $c^* = \arg\min_c d(\mathcal{X}, c)$ as the class label for the test sample.

Figure 6.2: The pseudo-code implementation of the R-UMLDA-A algorithm for tensor object recognition.

6.4 Experimental Study

This section investigates the various properties of the UMLDA algorithm. As usual, detailed results and comparisons on face and gait recognition will be presented in Chapter 7. The properties studied for UMLDA are: a) the effects of initialization, b) the effects of regularization, c) the effects of projection order, d) the convergence, e) the number of useful features, and f) the effects of aggregation.

Since different from MPCA and UMPCA, UMLDA is a supervised algorithm, the synthetic data sets used in the previous two chapters can not be used in this study. As discussed before, the behavior of supervised learning algorithms may be influenced significantly by the number of training samples per subject, denoted by L. Therefore, the UMLDA properties are studied on a subset of the PIE face database, which has a large number of samples for each subject. The study results obtained from the other face and gait databases are similar to the results on this PIE database.

In the PIE database used here, seven poses (C05, C07, C09, C27, C29, C37, C11) are chosen, with at most 45 degrees of pose variation and under the 21 illumination conditions (02 to 22). Thus, there are about 147 (7×21) samples per subject and there are a total number of 9,987 face images (with nine faces missing). All face images are normalized to 32×32 pixels, with 256 gray levels per pixel, as described in Section 3.2.3 (page 43). This database is randomly split into training and testing samples for the empirical study here.

In the implementation of R-UMLDA, in order to get better conditioned matrix for the inverse computation and to relax the limitation on the maximum number of uncorrelated features that can be extracted, a small term ($\varrho \cdot \mathbf{I}_{p-1}$) is added in computing the matrix inverse of $\left(\mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1}\right)$ in (6.15), with $\varrho = 10^{-3}$. In the implementation of the R-UMLDA-A, the maximum number of R-UMLDA to be aggregated is set to $A = 20$, and uniform initialization is used for $a = 1, 5, 9, 13, 17$ with corresponding $\gamma_a = 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ while random initialization is used for the rest values of

a.

6.4.1 The effects of initialization, regularization, and projection order

Figure 6.3 illustrates the effects of initialization and regularization on two face recognition experiments: one with $L = 2$ and one with $L = 20$, corresponding to the small sample size scenario and the scenario when a large number of samples (per subject) are available for training. The CRRs for various γs are depicted in Figs. 6.3(a) ($L = 2$) and 6.3(b) ($L = 20$) for the uniform initialization, and in Figs. 6.3(c) ($L = 2$) and 6.3(d) ($L = 20$) for the random initialization (averaged over 20 repeated trials). Figures 6.3(e) and 6.3(f) show the plots for the CRRs from eight repetitions of the random initialization with $\gamma = 10^{-3}$. They demonstrate that the recognition results are affected by initialization and different initialization results in different results. By comparing Fig. 6.3(f) against Fig. 6.3(e), it can be seen that the sensitivity to initialization is smaller for a larger L. Furthermore, by comparing Fig. 6.3(a) against Fig. 6.3(c), and Fig. 6.3(b) against Fig. 6.3(d), it is observed that the uniform initialization outperforms the random initialization for both a small L and a large L. Therefore, the uniform initialization is used when only one R-UMLDA feature extractor is employed, and in the following discussions on the convergence and the number of useful features, the results reported are obtained using the uniform initialization.

Besides the effects of initialization, the effects of regularization are also observed in Figs. 6.3(a), 6.3(b), 6.3(c), and 6.3(d). For a small L, the UMLDA with a strong regularization (larger γ) can outperform that without regularization ($\gamma = 0$), while for a large L, a too strong regularization may results in poorer performance, as observed in other regularization algorithms [92].

The effects of projection order are also studied. Similar to initialization, the projection order affects UMLDA in most cases but there is no guidance on its optimal determination

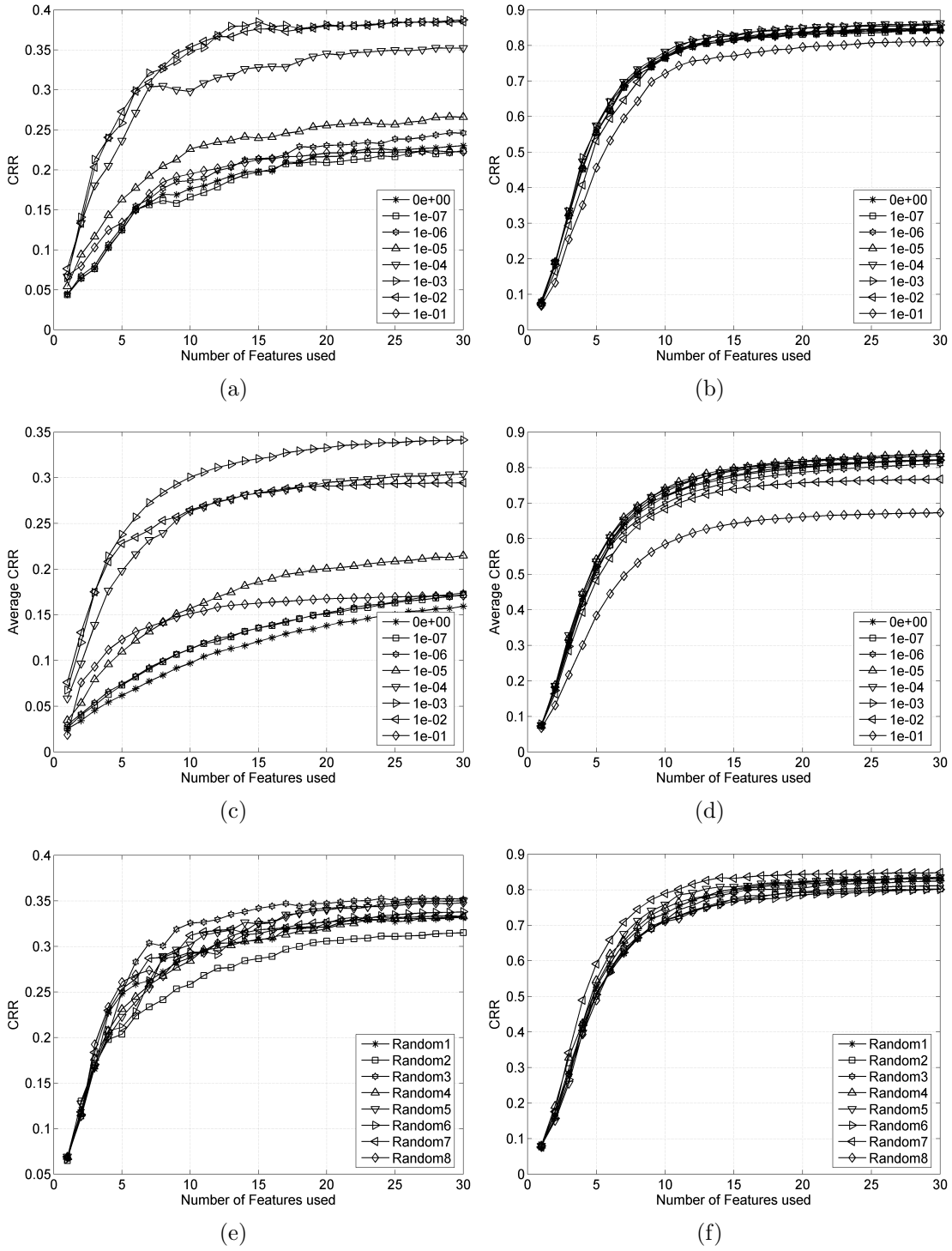


Figure 6.3: Illustration of the effects of initialization and regularization on the recognition performance of R-UMLDA: Uniform initialization with various γs for (a) $L = 2$ and (b) $L = 20$; Random initialization with various γs averaged over 20 repetitions for (c) $L = 2$ and (d) $L = 20$; Eight repetitions of random initialization with $\gamma = 10^{-3}$ for (e) $L = 2$ and (f) $L = 20$.

for recognition. If the computational cost in training is not a concern and it is possible to construct a validation set, all possible projection orders could be tested and the one with the best results on the validation set should be used. In this work, the projection order is fixed to be sequential from 1 to N, as in MPCA and UMPCA.

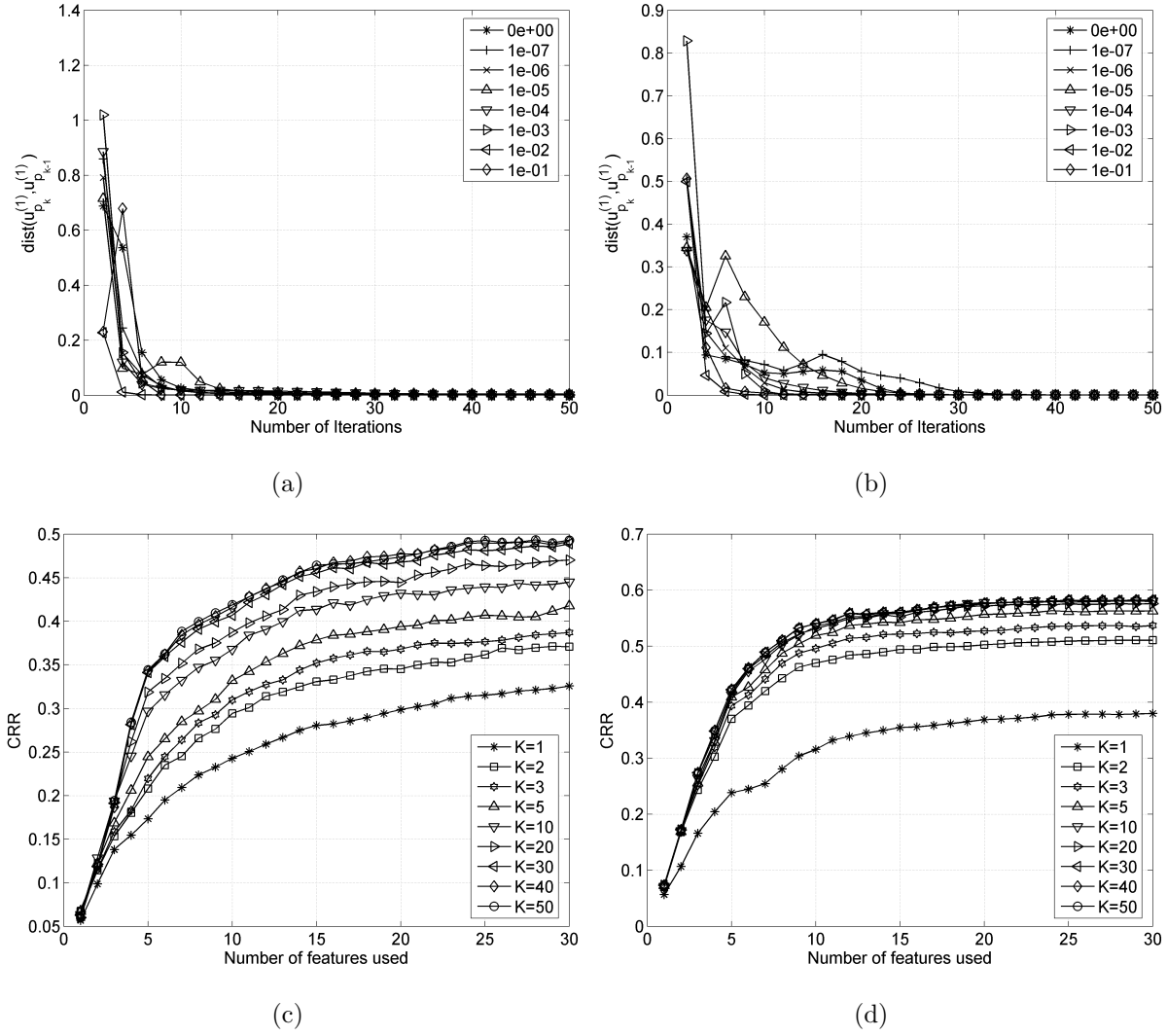


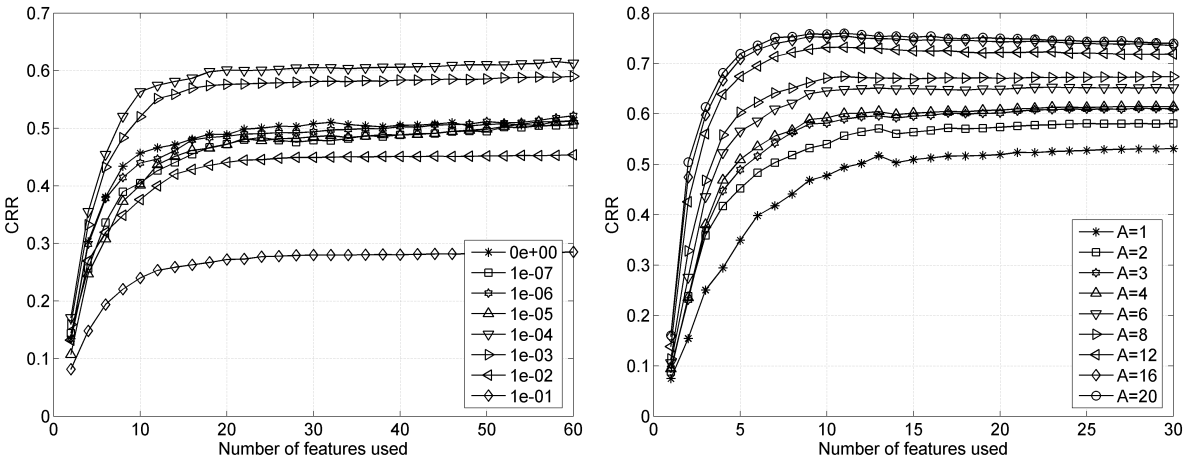
Figure 6.4: Illustration of the convergence of R-UMLDA for $L = 5$: the evolution of $dist\left(\mathbf{u}_{p_{(k)}}^{(1)}, \mathbf{u}_{p_{(k-1)}}^{(1)}\right)$ over 50 iterations for (a) $p = 1$ and (b) $p = 8$ with various γs (the legends); the CRRs for various Ks (the maximum number of iterations) for (c) $\gamma = 0$ and (d) $\gamma = 10^{-3}$.

6.4.2 Convergence

The convergence is illustrated in Fig. 6.4. Figures 6.4(a) and 6.4(b) depict two examples of the evolution of $dist\left(\mathbf{u}^{(1)}_{p_{(k)}}, \mathbf{u}^{(1)}_{p_{(k-1)}}\right)$ for $p = 1$ and $p = 8$, with various γs, for up to 50 iterations. As seen in the figure, in the worst scenarios, the projection vector converges around $k = 15$ for $p = 1$ and around $k = 30$ for $p = 8$. In addition, a stronger regularization (larger γ) is more likely to result in faster convergence. Furthermore, the recognition performance is examined for various Ks, as shown in Figs. 6.4(c) and 6.4(d) with $L = 5$ for $\gamma = 0$ and $\gamma = 10^{-3}$, respectively. It indicates that the first a few iterations improve the recognition performance the most, and more iterations afterwards give slow improvement in the recognition rate, especially for a larger γ. Therefore, K is set to a fixed number, $K = 10$, to terminate the iteration in practice. When computational efficiency is important, K can be further reduced to improve processing speed, while sacrificing some recognition performance.

6.4.3 The number of useful features and the effects of aggregation

R-UMLDA is limited in the number of extracted features (P) useful for recognition, as depicted in Fig. 6.5(a), where the CRRs are shown for up to 60 features for $L = 5$ and with various γs. In particular, the first a few features are very powerful, while beyond a certain number (e.g. 20), the performance varies very slowly with an increased P. Fortunately, from the study of the effects of initialization and regularization, it is found that different initialization or regularization produces different results (Fig. 6.3). Thus, the proposed aggregation scheme makes use of this property and combines differently initialized and regularized R-UMLDA recognizers to achieve enhanced results. At the same time, the problem of regularization parameter selection is alleviated. The results of aggregation are shown in Fig. 6.5(b) for $L = 5$ and up to 20 R-UMLDA recognizers



(a) (b)

Figure 6.5: Demonstration of (a) the recognition performance for $L = 5$ as P increases for various γs in R-UMLDA, and (b) the effectiveness of aggregation in R-UMLDA-A.

to be combined, by the R-UMLDA-A described at the beginning of this section. The figure demonstrates that the aggregation is an effective procedure and there are indeed complementary discriminative information from differently initialized and regularized R-UMLDA recognizers.

6.5 Summary

In this chapter, the UMLDA algorithm is proposed to extract discriminative features directly from tensorial data through solving a tensor-to-vector projection so that the traditional Fisher's discrimination criterion is maximized in each elementary projection, while the features extracted are constrained to be uncorrelated. In addition, an adaptive regularization factor is incorporated, resulting in the regularized UMLDA. The regularization aims to enhance the performance in practical applications where the input dimensionality is very high but the sample size per class is often limited, such as face or gait recognition [92, 81]. Furthermore, as different initialization or regularization of UMLDA results in different features, an aggregation scheme is proposed to combine several differently initialized and regularized UMLDA feature extractors at the matching

score level using the simple sum rule, so that the recognition performance is further improved. At the same time, the regularization parameter selection problem is alleviated in the proposed aggregation scheme. In the experimental section, various properties of the proposed algorithm are studied on a subset of the PIE face database. The presented experimental results explain the rational behind the proposed solution and algorithmic designs chosen, such as initialization, termination, regularization, and aggregation.

At this point, all the algorithms proposed in this research have been presented, with their respective properties studied. The three contributed solutions have addressed the three unexplored directions of multilinear subspace learning, as indicated by the shaded empty boxes in Figs. 3.7(a) and 3.7(b) (page 51). In the next chapter, these algorithms are evaluated on practical face and gait recognition problems through the comparison of recognition performance against competing solutions in the literature.

Chapter 7

Face and Gait Recognition Results

7.1 Introduction

The previous three chapters have introduced several new algorithms in both unsupervised and supervised multilinear subspace learning, with their properties studied. In this chapter, these algorithms will be applied to the face and gait recognition problems to evaluate their performance. Specifically, they will be tested on the face and gait databases described in Chapter 3 to investigate whether the proposed new algorithms can advance the current state of multilinear subspace learning and achieve better recognition results on face and gait recognition than existing subspace learning solutions. The following six sets of experiments have been designed for the performance evaluation:

1. The number of training samples available per class has significant effects on recognition performance. A subset of the PIE database is used to study the performance of the proposed and existing subspace learning algorithms under varying number of training samples per class.

2. Besides the number of training samples per class, the number of classes to be classified can affect the recognition performance of many algorithms. A subset of the FERET database is employed for the performance study of subspace learning

algorithms under varying number of classes.

3. As shown in Corollary 5.1 (page 103), UMPCA can only extract a limited number of uncorrelated features. Thus, it is expected to perform poorly compared to other algorithms in the two sets of experiments above. However, for the case of unsupervised learning in low-dimensional subspace, UMPCA can be particularly effective. Thus, a specific study is carried out on another subset of the FERET database.

4. The USF gait database has seven probes captured under different conditions and it is used to test the gait recognition performance of the proposed and existing subspace learning algorithms under conditions with various difficulties.

5. Many state-of-the-art gait recognition algorithms in the literature have more sophisticated preprocessing and matching algorithms than those described in Chapter 3. Therefore, it is worthwhile to compare the MPCA-based algorithms, which show good performance in the first set of gait recognition experiments above, against those state-of-the-art gait recognition algorithms.

6. The boosting framework has been studied on the face recognition problem, with promising results reported [93]. However, there is no similar investigation on the gait recognition problem yet. The final set of experiments studies whether the MPCA+boosting algorithm can be beneficial in gait recognition.

This chapter proceeds with a discussion in Section 7.2 on the algorithms to be compared and their respective settings in the experiments. Then, a comprehensive comparison of all these algorithms are carried out on both face and gait recognition problems in Sections 7.3 and 7.4. Section 7.3 presents the recognition results on both the PIE and FERET face databases and Section 7.4 presents the results on the USF gait database. Next, Section 7.5 discusses observations from the experimental evaluation. Finally, this chapter is summarized in Section 7.6.

Table 7.1: List of unsupervised subspace learning algorithms to be compared.

Acronym	Full name	Mode
PCA	principal component analysis [47, 128]	Linear
2DPCA	two-dimensional PCA [153]	Multilinear
CSA	concurrent subspaces analysis [146]	Multilinear
TROD	tensor rank-one decomposition [116]	Multilinear
MPCA	multilinear PCA [Chapter 4]	Multilinear
UMPCA	uncorrelated MPCA [Chapter 5]	Multilinear

Table 7.2: List of supervised subspace learning algorithms to be compared.

Acronym	Full name	Mode
PCA+LDA	PCA+linear discriminant analysis [3]	Linear
ULDA	uncorrelated linear discriminant analysis [154]	Linear
R-JD-LDA	regularized version of the revised direct LDA [92, 90]	Linear
DATER	discriminant analysis with tensor representation [150]	Multilinear
GTDA	general tensor discriminant analysis [124]	Multilinear
TR1DA	tensor rank-one discriminant analysis [142, 123]	Multilinear
MPCA-S	MPCA with discriminative feature selection [Chapter 4]	Multilinear
MPCA+LDA	MPCA-S+linear discriminant analysis [Chapter 4]	Multilinear
R-UMLDA	regularized uncorrelated multilinear discriminant analysis [Chapter 6]	Multilinear
R-UMLDA-A	R-UMLDA with aggregation [Chapter 6]	Multilinear

7.2 Algorithms and Their Settings

The new multilinear subspace learning algorithms proposed in this dissertation are compared against other linear or multilinear subspace learning algorithms in the literature on the problems of face and gait recognition. The unsupervised and supervised subspace learning algorithms to be compared are listed in Tables 7.1 and 7.2, respectively. It should be noted that in Table 7.2, the ULDA algorithm compared here is different from the ULDA in [45] so it is different from the classical LDA. The MPCA-S algorithm is

MPCA with discriminative feature selection described in Section 4.4 (page 78). CSA is implemented as MPCA without centering of the data. For feature extraction using CSA and MPCA, including MPCA-S and MPCA+LDA, the Q-based method described in Section 4.3.6 (page 74) is used to determine the subspace dimensionality in projection, with a fixed $Q = 0.97$ (97%). In addition, 2DPCA is only applied to the face recognition problem since it cannot handle the third-order tensors in gait recognition.

As discussed in Section 3.1 (page 37), the recognition performance is evaluated by the identification rate calculated through similarity measurement between feature vectors. However, among the algorithms considered here, 2DPCA, CSA, MPCA, DATER, and GTDA produce tensorial features and they need to be vectorized for classification. Hence, for the unsupervised methods, 2DPCA, CSA, and MPCA, each entry in the projected tensorial features are viewed as an individual feature and the corresponding total scatter as defined in Definition 2.4 (page 32) is calculated. The tensorial features produced by these methods are then arranged into a vector in descending total scatter. For the supervised methods, DATER and GTDA, the projection is obtained with $P_n = I_n$ for $n = 1, ..., N$ and then the tensor-to-tensor projection is viewed as $\prod_{n=1}^{N} I_n$ elementary multilinear projections. As in Section 4.4 (page 78), the discriminability of each such elementary multilinear projection is calculated on the training set and the tensorial feature is arranged into a feature vector in descending discriminability.

All the iterative algorithms are terminated by setting the maximum number of iterations K for fair comparison and computational concerns. Since CSA and MPCA have very good convergence performance, K is set to 1. For all the other algorithms (TROD, UMPCA, DATER, GTDA, TR1DA, R-UMLDA), K is set to 10. For all the algorithms, up to 600 features, while not exceeding the maximum number of features, are tested, unless stated otherwise. For instance, LDA, ULDA, and R-JD-LDA algorithms can not extract more than $(C - 1)$ features.

For the recognition experiments of only one R-UMLDA, the uniform initialization is

used and γ is empirically set to 10^{-3}, with up to 30 features tested. For R-UMLDA-A, up to 20 differently initialized and regularized versions of the UMLDA feature extractors are combined with each producing up to 30 features, also resulting in a total number of 600 features. The aggregation parameter settings are the same as those in Section 6.4 (page 130).

For the TR1DA algorithm described in Section 3.4.4 (page 57), the tuning parameter ζ needs to be set heuristically. Several values of ζ for each L are tested on the PIE database. The best ζ used for each range of L is: $\zeta = 2$ for $L \leq 7$, $\zeta = 0.8$ for $8 \leq L \leq 15$, and $\zeta = 0.6$ for $L \geq 16$. For the GTDA algorithm, ζ is set following the procedure suggested in [124]. For R-JD-LDA, the default maximum number ($\approx 0.8 \cdot (C-1)$) of features and a regularization parameter of 0.001 originally suggested by the authors of [92,90] are used.

For face recognition performance evaluation, the CRR, i.e., the rank 1 identification rate, is reported. Since gait is a more difficult biometric to recognize, both the rank 1 and rank 5 identification rates are reported for gait recognition performance evaluation. In calculating the identification rates, the similarity between feature vectors measured using the L1, L2, and angle distance measures in Table 3.1 (page 40) are tested for each algorithm. The one resulting in the best performance is reported here, unless stated otherwise.

For each best-performing distance measure, the best recognition results reported are obtained by varying the number of features used and the number of R-UMDLA recognizers aggregated for the R-UMLDA-A algorithm. For the PCA+LDA and MPCA+LDA algorithms, the dimensionality of the feature vectors for input to LDA can affect the performance. Therefore, 18 values of the PCA/MPCA dimensions for input to LDA are tested, sampled with equal spacing from 80 to 600, and the best results obtained will be compared against other algorithms. For all the other algorithms, the best results are obtained by only varying the number of features used. For fair comparison, there is no further fine tuning of other parameters (such as the regularization parameter for the R-

JD-LDA) for optimal performance on the testing data, including the proposed methods. For better viewing, the top two recognition results in each experiment are shown in bold when reporting them in tables.

7.3 Face Recognition Results

In the face recognition experiments, face images are input directly as second-order tensors to the multilinear algorithms, while for the linear algorithms, they are converted to vectors as input. For each subject in a face recognition experiment, L samples are randomly selected for training and the rest are used for testing. Accordingly, in presenting the recognition results, the mean and standard deviation (Std) over 20 random splits are reported, unless stated otherwise. In the following, three sets of face recognition experiments are presented.

7.3.1 Face recognition results on the PIE database

The first set of experiments is on the subset from the PIE database used in Section 6.4 (page 130), with seven poses (C05, C07, C09, C27, C29, C37, C11) of at most 45 degrees of pose variation and under the 21 illumination conditions (02 to 22). This subset contains 9,987 face images, around 147 samples per subject. All face images are preprocessed to 32×32 pixels, with 256 gray levels per pixel.

In order to study the recognition performance with different Ls, nine face recognition experiments are performed on this PIE database with $L = 2, 3, 4, 5, 6, 8, 10, 20, 40$. The top CRRs are listed in Table 7.3, where the MPCA+LDA and the R-UMLDA-A algorithms give the best overall performance. The detailed results for $L = 2, 4, 6, 10, 20$, and 40 by the unsupervised and supervised algorithms are depicted in Figs. 7.1 and 7.2, respectively, where the horizontal axis is shown in log scale. From Fig. 7.1, it is observed that although UMPCA performs poorly due to the limited number of features

Table 7.3: Face recognition results on the PIE database: the top CRRs (Mean±Std%) for various Ls.

L	2	3	4	5	6	8	10	20	40
PCA	28.5±0.8	36.5±0.9	43.7±1.1	49.2±0.8	53.7±0.9	61.7±0.8	67.6±0.9	84.4±0.7	96.1±0.5
2DPCA	21.6±0.7	27.7±1.0	33.1±0.9	37.6±0.6	41.4±0.5	48.7±0.7	54.5±0.9	73.5±0.8	90.1±0.5
CSA	20.3±0.9	26.0±1.1	30.9±0.7	34.7±0.7	38.2±0.7	44.6±0.8	49.5±0.8	66.3±0.7	83.2±0.6
TROD	21.8±0.9	28.3±1.2	34.0±1.1	38.5±0.9	42.6±0.8	50.1±0.9	55.7±0.9	74.6±0.9	90.5±0.6
MPCA	25.3±0.8	32.5±0.9	38.7±1.0	43.7±0.7	47.9±0.8	55.4±0.8	61.4±0.9	79.3±0.7	93.6±0.5
UMPCA	21.4±1.7	27.2±2.3	31.3±1.4	35.4±2.1	39.4±2.0	45.9±1.7	51.5±2.6	67.7±1.8	84.2±1.6
PCA+LDA	**48.0**±1.7	60.1±1.7	67.6±2.1	72.2±1.3	76.4±1.4	83.0±1.2	87.2±0.9	86.4±1.1	**98.6**±0.2
ULDA	44.1±1.3	55.7±1.3	63.4±1.4	67.8±1.1	70.7±0.9	74.5±1.0	75.2±1.1	83.6±1.0	97.7±0.3
R-JD-LDA	42.7±2.0	58.9±1.7	66.5±1.9	72.1±1.2	75.7±1.4	81.1±1.0	85.1±0.9	93.5±0.6	98.0±0.3
DATER	46.3±2.3	**61.0**±2.2	**68.7**±1.5	**73.2**±1.8	77.0±1.4	82.9±0.7	85.8±0.8	93.3±0.6	98.2±0.4
GTDA	40.1±2.0	50.4±2.0	57.3±1.4	61.8±1.5	65.7±1.4	71.9±1.1	76.6±0.9	88.3±0.6	96.5±0.5
TR1DA	33.5±2.9	52.6±1.4	63.7±1.8	69.2±1.5	72.8±1.7	76.1±1.2	81.0±1.1	90.0±0.6	95.7±0.4
MPCA-S	36.9±4.7	48.3±2.4	54.9±1.5	59.4±1.4	63.5±1.5	70.3±0.9	74.8±1.0	87.3±0.5	96.0±0.4
MPCA+LDA	**50.4**±1.5	**61.2**±1.7	66.7±1.7	**73.2**±2.3	**78.7**±1.3	**85.4**±0.8	**89.3**±0.8	**96.6**±0.4	**99.3**±0.1
R-UMLDA	40.3±1.7	50.9±1.7	58.3±1.3	63.4±1.5	67.0±1.1	72.7±1.0	76.7±1.4	87.7±0.7	95.1±0.3
R-UMLDA-A	45.2±1.8	58.6±1.7	**69.0**±2.1	**74.9**±1.2	**79.4**±1.2	**85.0**±1.0	**88.7**±0.9	**95.2**±0.4	**98.6**±0.3

that it can extract, this algorithm outperforms all the other unsupervised algorithms in the low-dimensional subspace (for $P = 1, ..., 32$). It is also noted from Fig. 7.2 that for $L = 10, 20$, the first a few features extracted by R-UMLDA are the most powerful features in recognition. For all values of L in Fig. 7.2, R-UMLDA outperforms most algorithms except PCA+LDA and MPCA+LDA in low-dimensional subspace. Among the unsupervised algorithms, PCA gives the best performance. It is worth noting that MPCA greatly outperforms CSA for all Ls, indicating that centering does have a positive impact on recognition. MPCA also outperforms 2DPCA and TROD for all Ls. Furthermore, MPCA-S outperforms MPCA significantly, , showing the effectiveness of the proposed discriminative feature selection procedure for MPCA.

Figure 7.3 shows the typical recognition results of PCA+LDA and MPCA+LDA algorithms for $L = 2, 6, 40$, under various PCA/MPCA dimensions before LDA. The

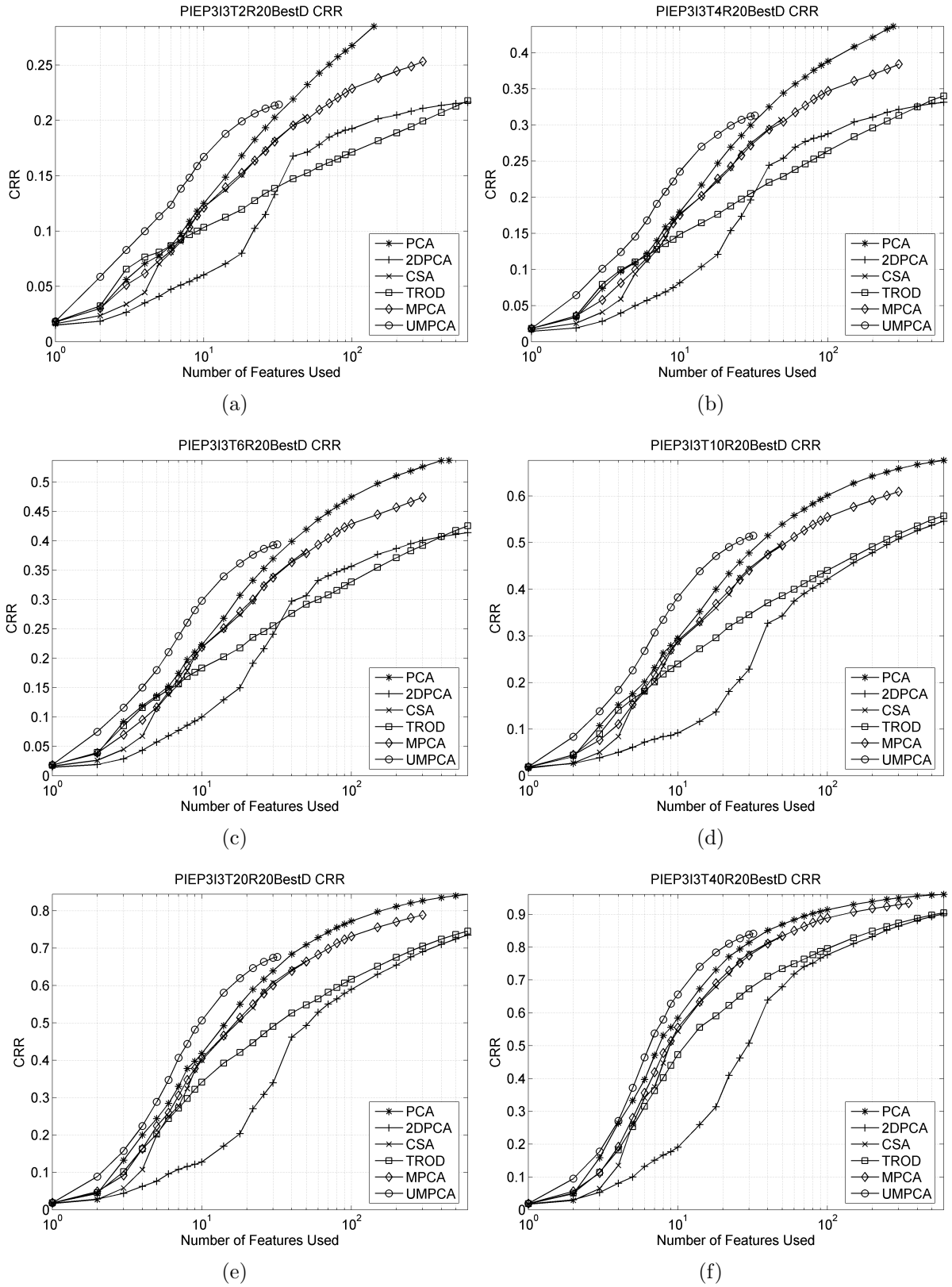


Figure 7.1: Face recognition results by unsupervised learning on the PIE database: CRR against the number of features used for (a) $L = 2$, (b) $L = 4$, (c) $L = 6$, (d) $L = 10$, (e) $L = 20$, and (f) $L = 40$.

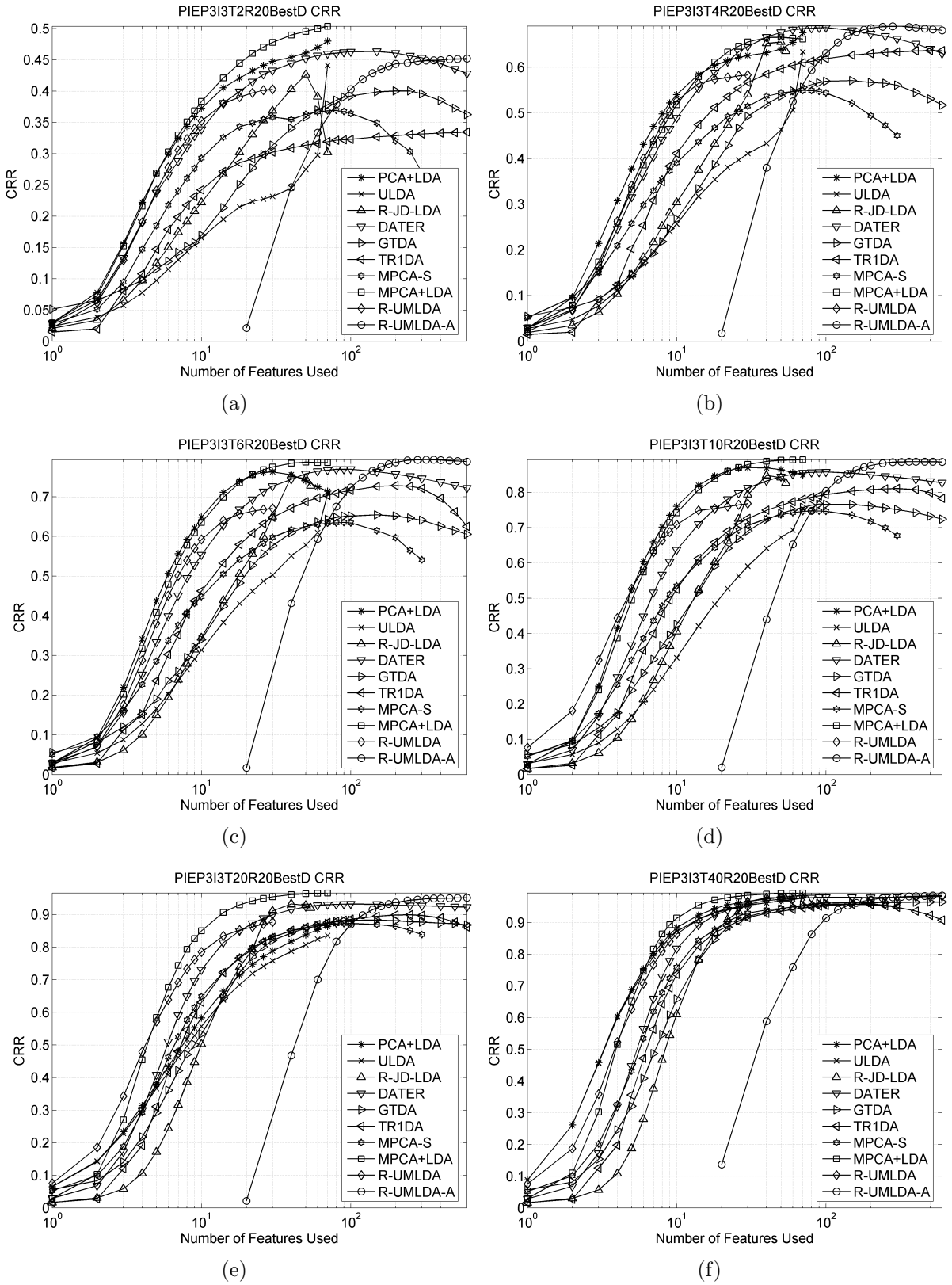


Figure 7.2: Face recognition results by supervised learning on the PIE database: CRR against the number of features used for (a) $L = 2$, (b) $L = 4$, (c) $L = 6$, (d) $L = 10$, (e) $L = 20$, and (f) $L = 40$.

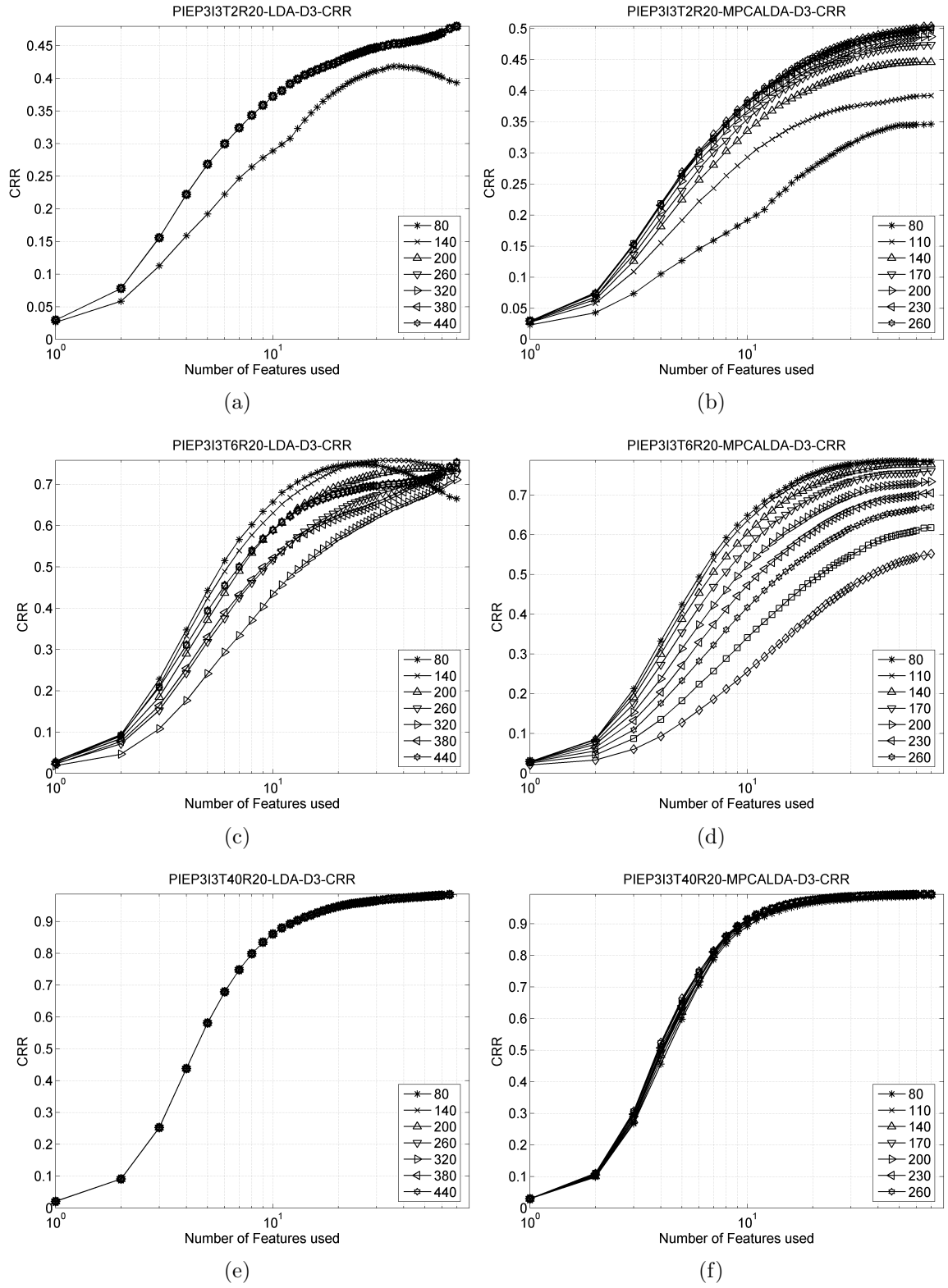


Figure 7.3: The sensitivity of the face recognition results on: the PCA dimensionality in the PCA+LDA algorithm for (a) $L = 2$, (c) $L = 6$, and (e) $L = 40$; the MPCA dimensionality in the MPCA+LDA algorithm for (b) $L = 2$, (d) $L = 6$, and (f) $L = 40$, tested on the PIE database with the angle distance measure. The seven legends indicate the seven PCA/MPCA dimensions tested.

horizontal axis is shown in log scale in this figure. It can be seen that the performance of the two algorithms is sensitive to the PCA/MPCA dimensionality before LDA except when there are a large number of samples per class for training. Therefore, they require the determination of appropriate PCA/MPCA dimensionality in practice for best performance.

In addition, it should be noted that in each experiment, if the regularization parameter for R-JD-LDA and R-UMLDA, and the range of γ for R-UMLDA-A are tuned, improved performance can be obtained since stronger regularization results in better performance for a small L and weaker regularization is better for a larger L [92]. Nonetheless, with fixed range of γ, R-UMLDA-A still outperforms all the other algorithms except MPCA+LDA for most of the L values ranging from 2 to 40.

7.3.2 Face recognition results by supervised subspace learning on the FERET database

It is pointed out in [93] that the learning capacity of any LDA-like algorithm is directly proportional to L, and reciprocally proportional to C. Thus, the recognition performance with different Cs is evaluated in this subsection. A subset is selected from the FERET database for this purpose and it is composed of those subjects with each having at least six images with at most 45 degrees of pose variation, resulting in 2,803 face images from 335 subjects. Face images from this FERET database are also preprocessed to 32×32 pixels, with 256 gray levels per pixel, as described in Section 3.2.3 (page 43). Four experiments are carried out on this database with $C = 80, 160, 240, 320$ and fixed $L = 4$ so that no more than half of the face images are used for training. The numbers of training and testing faces for each experiment are detailed in Table 7.4.

The CRRs for all the unsupervised subspace learning algorithms are below 50% in this set of experiments so they are not reported here. Table 7.5 lists the top CRRs for supervised subspace learning algorithms, where MPCA+LDA and R-UMLDA-A outper-

Table 7.4: The four experiments testing performance for different number of classes (C) on the FERET database.

C	number of training faces	number of testing faces
80	320	825
160	640	1113
240	960	1273
320	1280	1433

Table 7.5: Face recognition results by supervised subspace learning algorithms on the FERET database: the top CRRs (Mean±Std%) for various Cs.

C	80	160	240	320
PCA+LDA	69.7± 2.2	71.8±1.3	73.0±1.0	73.7±1.5
ULDA	58.9± 1.5	40.2±1.3	10.8±0.8	21.0±0.9
R-JD-LDA	70.7± 1.9	72.5±1.5	70.2±0.9	66.4±1.0
DATER	71.7± 1.5	69.3±2.0	64.7±1.0	63.2 ±1.5
GTDA	63.3± 2.0	61.1±2.7	58.3±1.6	55.8±1.7
TR1DA	71.5± 1.5	69.1±2.2	63.3±1.6	60.3±2.0
MPCA-S	59.2± 2.7	54.9±2.8	55.1±2.0	53.0±1.5
MPCA+LDA	**74.0**± 1.7	**77.1**±1.1	**77.2**±1.1	**77.9**±1.5
R-UMLDA	59.6± 1.8	58.0±1.6	56.3±1.6	56.9±1.4
R-UMLDA-A	**75.0**± 1.8	**75.7**±1.7	**74.1**±1.2	**73.8**±1.4

form all the other methods in all cases. Moreover, it can be observed that the recognition performance of PCA+LDA, MPCA+LDA, and R-UMLDA-A are just slightly affected by C. PCA+LDA and MPCA+LDA even get better performance with a larger C, indicating their capability in handling large number of classes. In contrast, ULDA and TR1DA are affected more by C, with decreased recognition rates as C increases. Detailed recognition results are shown in Fig. 7.4 with the horizontal axis in log scale. It is observed that in most cases, the first a few (around 7) features extracted by R-UMLDA are the most discriminative ones.

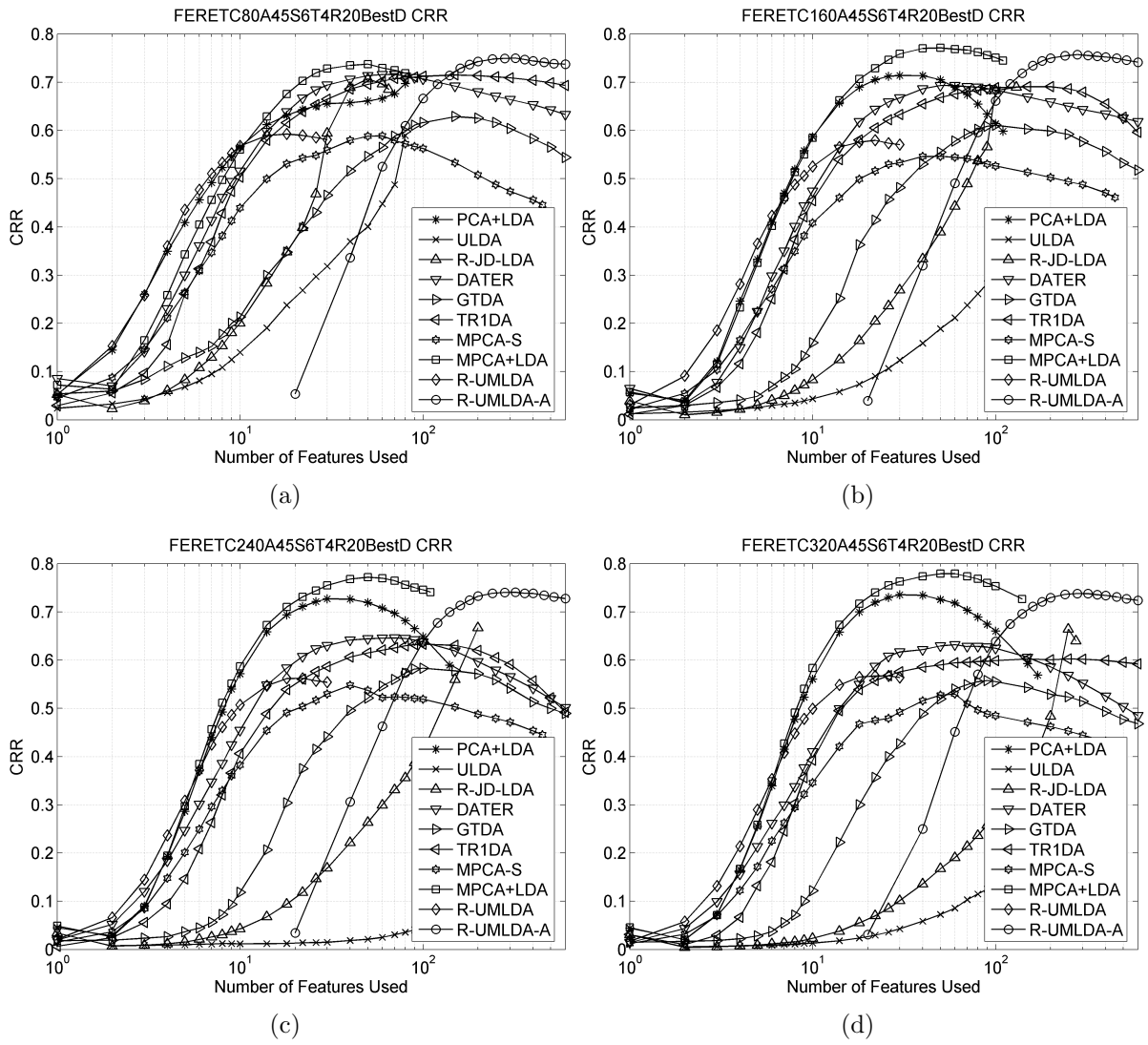
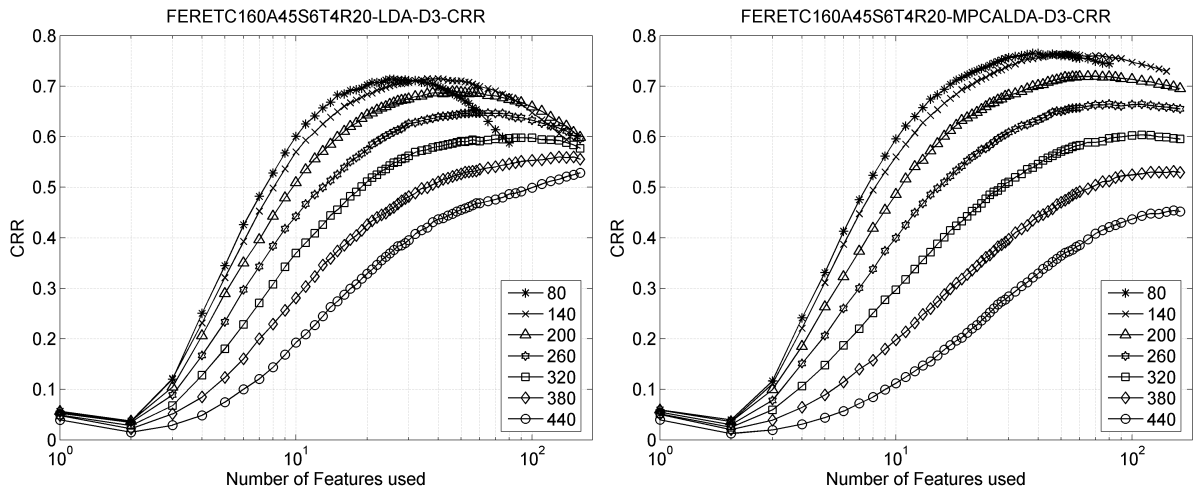


Figure 7.4: Face recognition results by supervised subspace learning algorithms on the FERET database: CRR against the number of features used for (a) $C = 80$, (b) $C = 160$, (c) $C = 240$, and (d) $C = 320$.

For PCA+LDA and MPCA+LDA, it is noted again that their performance is sensitive to the dimensionality of PCA or MPCA feature vectors to be fed into LDA. Figure 7.5 shows some typical face recognition results of the PCA+LDA and MPCA+LDA algorithms with various PCA/MPCA dimensions before LDA, obtained on the FERET database with $C = 160$ and the angle distance measure. The horizontal axis is again shown in log scale. As seen from the figure again, it is important for these two algorithms to choose the appropriate dimensionality before LDA.



(a) (b)

Figure 7.5: The sensitivity of the face recognition results on the PCA/MPCA dimensionality in (a) PCA+LDA and (b) MPCA+LDA, tested on the FERET database with $C = 160$ and the angle distance measure. The seven legends indicate the seven PCA/MPCA dimensions tested.

7.3.3 Face recognition by unsupervised learning in low-dimensional subspace

In the previous two sets of face recognition experiments, it has been observed that the unsupervised subspace learning algorithms have poor performance in most cases, compared with the supervised counterparts, partly due to their unsupervised nature and the difficulty of the databases chosen. In addition, it is also noted that UMPCA can only produce a limited number of features, as analyzed in Corollary 5.1 (page 103), so its performance on low resolution faces is poor in turn. On the other hand, since no class specific information is required in the learning process, the unsupervised subspace learning methods have wider applications. In particular, in the so-called one training sample case important in practice [138], an extreme small sample size scenario where only one sample per class is available for training (i.e., $L = 1$), the supervised subspace learning algorithms studied in this dissertation cannot be applied since it is impossible to measure the within-class scatter with only one sample per class available. In contrast, the

unsupervised subspace learning methods are still applicable even in this difficult scenario.

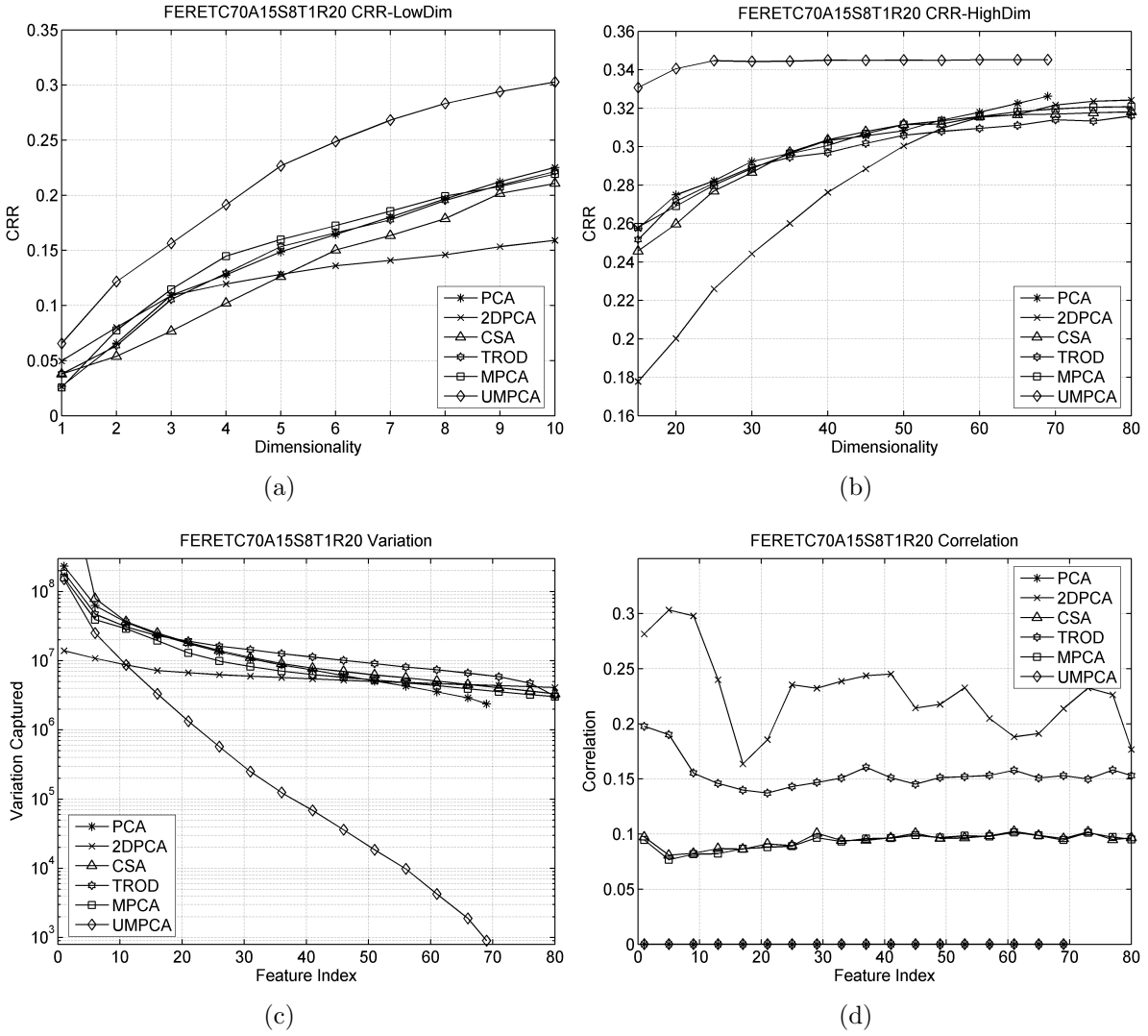


Figure 7.6: Detailed face recognition results by unsupervised subspace learning algorithms on the FERET database for $L = 1$: (a) performance curves for the low-dimensional case, (b) performance curves for the high-dimensional case, (c) the variation captured by individual features, and (d) the correlation among features.

In this subsection, a higher resolution and less challenging face database is constructed to study the recognition performance of the unsupervised subspace learning algorithms, particularly in low-dimensional subspace. Another subset of the FERET database is selected to consist of those subjects with each having at least eight images with at most 15 degrees of pose variation, resulting in 721 face images from 70 subjects. All face

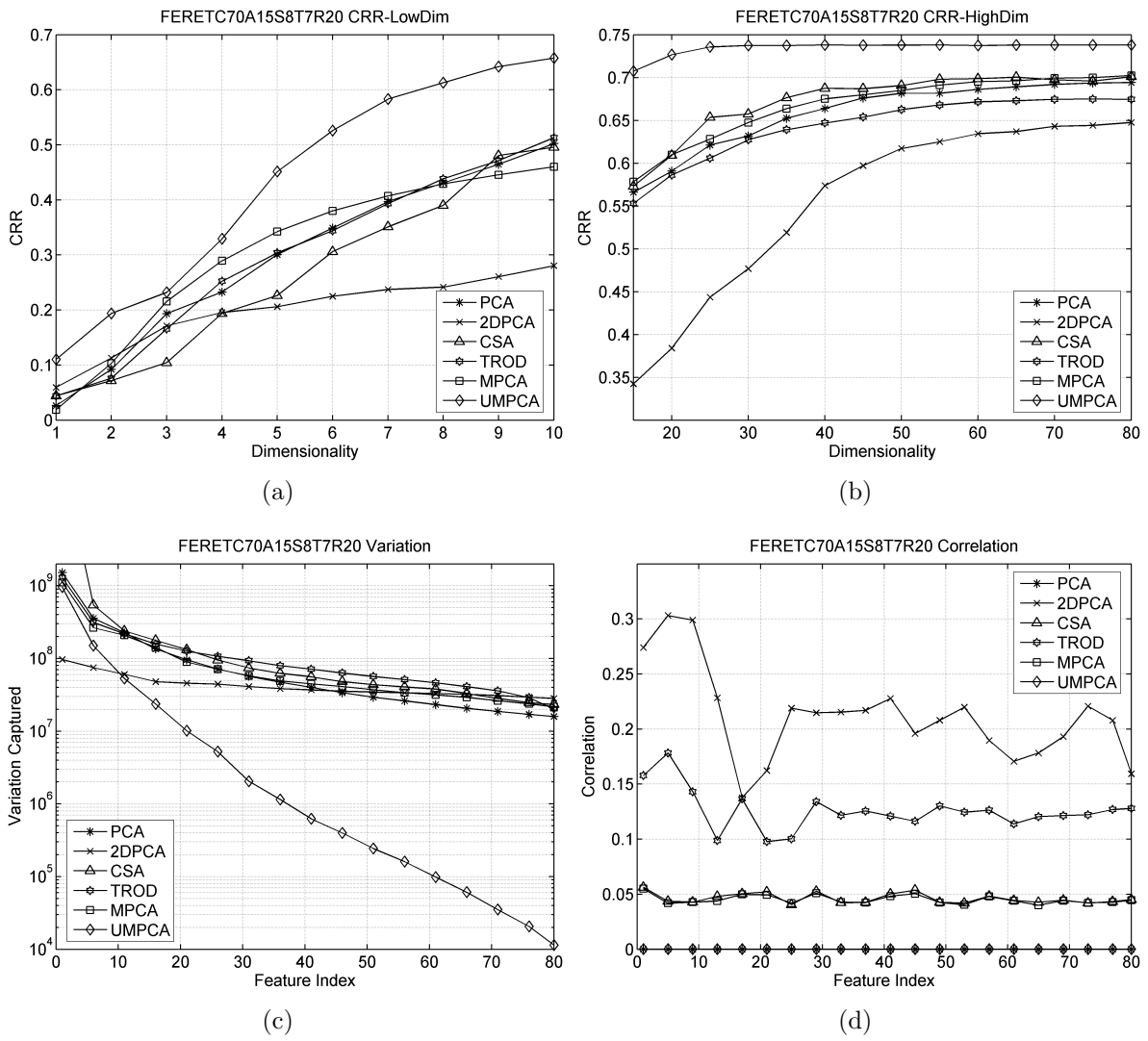


Figure 7.7: Detailed face recognition results by unsupervised subspace learning algorithms on the FERET database for $L = 7$: (a) performance curves for the low-dimensional case, (b) performance curves for the high-dimensional case, (c) the variation captured by individual features, and (d) the correlation among features.

images are preprocessed as usual and normalized to 80×80 pixels, with 256 gray levels per pixel.

For the unsupervised subspace learning algorithms, the extracted features are all arranged in descending variation captured, (measured by respective total scatter). In this set of experiments, only the L2 (Euclidean) distance measure is tested. For each subject in a face recognition experiment, $L(= 1, 2, 3, 4, 5, 6, 7)$ samples are randomly

selected for training and the rest are used for testing.

Figures 7.6 and 7.7 show the detailed results for $L = 1$ and $L = 7$, respectively. It should be noted that for PCA and UMPCA, there are at most 69 (uncorrelated) features when $L = 1$ since there are only 70 faces for training. As mentioned above, $L = 1$ is the one training sample (per class) case, and $L = 7$ is the maximum number of training samples that can be used in this set of experiments. Figures 7.6(a) and 7.7(a) plot the CRRs against P, the dimensionality of the subspace, for $P = 1, ..., 10$, and Figs 7.6(b) and 7.7(b) plot those for P ranging from 15 to 80. From the figures, UMPCA outperforms the other five methods in both cases and across Ps, indicating that the uncorrelated features extracted directly from the tensorial face data are effective in classification. The figures also show that for UMPCA, the recognition rate saturates around $P = 30$, which can be explained by observing the variation captured by individual features as shown in Figs. 7.6(c) and 7.7(c) (in log scale). These figures show that the variation captured by UMPCA is considerably lower than those captured by the other methods, due to its constraints of zero-correlation and the tensor-to-vector projection. Despite capturing lower variation, UMPCA is superior in the recognition task performed. Nonetheless, when the variation captured is too low, those corresponding features are no longer descriptive enough to contribute in classification, leading to the saturation.

In addition, the average correlations of individual features with all the other features are plotted in Figs. 7.6(d) and 7.7(d). As supported by theoretical derivation, features extracted by PCA and UMPCA are uncorrelated. In contrast, the features extracted by all the other methods are correlated, with those extracted by 2DPCA and TROD have much higher correlation on average, which could be partly the reason of their poorer performance.

The recognition results for $P = 1, 5, 10, 20, 50, 80$ are listed in Table 7.6 for $L = 2, 3, 4, 5, 6$. From the table, UMPCA achieves the best recognition results in all cases reported. In particular, for smaller P $(1, 5, 10, 20)$, UMPCA outperforms the other methods

Table 7.6: Face recognition results by unsupervised subspace learning algorithms on a less challenging FERET database: the CRRs (Mean±Std%) for various Ls and Ps.

L	P	1	5	10	20	50	80
2	PCA	2.8±0.7	20.3±1.5	**31.5**±2.3	**38.4**±2.4	43.1±2.8	44.4±2.6
	2DPCA	**5.3**±0.9	15.9±1.9	19.5±2.1	26.0±3.2	40.4±2.8	44.2±2.4
	CSA	3.5±0.7	15.5±1.0	29.2±1.9	36.3±2.2	43.8±2.6	45.0±2.8
	TROD	3.5±0.8	19.9±3.4	30.1±2.1	37.7±2.4	42.3±2.1	43.8±2.5
	MPCA	2.6±0.6	**21.4**±1.5	28.4±1.8	38.2±2.2	**43.9**±2.7	**45.2**±2.7
	UMPCA	**7.9**±1.5	**30.0**±5.2	**41.7**±5.6	**46.1**±6.0	**46.7**±6.3	**46.7**±6.3
3	PCA	2.7±0.6	24.5±1.9	**38.0**±2.2	**46.3**±2.2	51.8±2.7	53.0±2.5
	2DPCA	**5.1**±0.9	17.3±1.5	22.3±1.8	30.5±2.7	47.4±2.6	51.8±2.2
	CSA	3.9±0.8	17.3±1.6	36.4±1.5	44.4±1.9	52.1±2.6	53.5±2.8
	TROD	3.9±0.7	23.2±3.3	36.4±2.3	45.1±2.4	50.5±2.7	52.2±2.6
	MPCA	2.4±0.6	**25.8**±1.6	34.9±2.3	45.8±2.2	**52.3**±2.7	**53.9**±2.8
	UMPCA	**7.5**±1.0	**35.3**±3.8	**49.7**±3.6	**56.0**±4.0	**56.7**±4.3	**56.6**±4.3
4	PCA	2.8±0.7	26.7±2.4	**42.5**±2.3	50.2±1.8	57.8±2.2	58.8±2.4
	2DPCA	**5.4**±0.6	18.3±1.1	24.3±1.7	34.1±4.3	51.7±2.5	56.4±2.5
	CSA	3.7±1.0	19.0±1.4	41.2±2.4	50.2±2.1	**58.4**±2.8	**59.6**±2.5
	TROD	3.8±0.9	25.3±2.6	42.2±3.1	50.0±2.6	55.6±2.1	57.6±2.4
	MPCA	2.3±0.6	**29.5**±2.3	40.4±2.4	**51.2**±2.5	58.3±2.5	**59.6**±2.3
	UMPCA	**8.1**±1.3	**40.1**±3.8	**56.9**±3.0	**63.3**±3.3	**64.0**±3.6	**64.0**±3.6
5	PCA	2.8±0.8	29.2±1.9	**47.0**±1.7	55.5±2.0	63.6±1.5	64.8±1.5
	2DPCA	**5.6**±1.2	19.9±1.6	26.4±2.4	36.4±3.5	57.0±2.5	61.6±2.3
	CSA	4.2±1.1	20.7±1.9	46.0±2.2	56.1±2.5	**64.8**±2.1	65.6±1.7
	TROD	4.2±1.1	28.9±3.0	46.7±2.9	55.6±2.1	61.6±1.9	63.7±1.8
	MPCA	2.6±0.7	**32.6**±2.1	43.0±2.6	**57.0**±2.2	64.4±2.1	**65.9**±1.7
	UMPCA	**8.5**±1.6	**42.5**±4.5	**61.0**±5.2	**67.7**±5.0	**68.7**±5.1	**68.7**±5.1
6	PCA	2.6±0.8	30.0±2.0	49.6±2.9	58.3±2.5	66.6±2.2	67.9±2.3
	2DPCA	**5.4**±1.4	20.9±1.9	27.9±2.7	38.3±2.9	58.1±1.9	63.2±2.4
	CSA	4.0±0.8	22.4±1.9	49.1±2.4	59.5±2.7	**68.0**±2.6	69.0±2.4
	TROD	4.3±0.7	28.5±2.8	**50.3**±2.3	58.7±2.7	64.8±2.3	66.7±2.0
	MPCA	2.2±1.0	**34.2**±2.4	46.4±2.6	**60.5**±2.6	67.5±2.5	**69.4**±2.3
	UMPCA	**9.0**±1.2	**44.5**±4.2	**63.1**±4.5	**70.4**±4.8	**71.4**±4.9	**71.3**±4.9

significantly, demonstrating its superior capability in classifying faces in low-dimensional spaces.

7.4 Gait Recognition Results

The experiments in the previous section demonstrate the effectiveness of the proposed solutions on second-order tensor objects, under various scenarios. In this section, the performance of the proposed algorithms are tested on third-order tensorial gait objects. In the gait recognition experiments, gait samples are input directly as third-order tensors to the multilinear algorithms, while for the linear algorithms, they are converted to vectors for input. The standard testing procedures described in Section 3.3 (page 44) are followed. Since R-UMLDA-A involves random initialization, the results obtained from this algorithm are reported in the mean and standard deviation over 20 repeated experiments.

7.4.1 Gait recognition results by subspace learning algorithms

The gait recognition experiments are carried out on the USF gait database V.1.7 described in Section 3.3 (page 44) to study the performance of the subspace learning algorithms on probes with varying difficulty. The original resolution $128 \times 88 \times 20$ results in vectors of 225280×1, which makes most linear subspace learning algorithms infeasible. Therefore, in this set of experiments, each normalized gait sample is downsampled to a tensor of $32 \times 22 \times 10$ so that all linear subspace learning algorithms can be applied. In addition, besides the rank 1 and rank 5 identification rates based on gait sequence matching, the CRRs for individual gait samples are also reported.

Tables 7.7, 7.8, and 7.9 present the CRRs for individual gait samples, the rank 1 and rank 5 identification rates for gait sequences, respectively. The average for probes A, B, and C (the easier probes) as well as the average over all the seven probes are also reported.

In addition, Figs. 7.8 and 7.9 plot the detailed average recognition performance for supervised and unsupervised subspace learning algorithms, respectively. The horizontal axis is shown in log scale in these figures.

Table 7.7: Gait recognition results on the $32 \times 22 \times 10$ USF gait database V.1.7: the CRR (%) for individual samples. MeanABC is the average over probes A, B, and C and MeanAll is the average over all seven probes.

Probe	A	B	C	D	E	F	G	MeanABC	MeanAll
PCA	54.2	46.6	31.0	18.9	17.5	12.0	12.5	43.6	27.0
CSA	47.5	43.0	26.0	18.9	14.7	11.1	11.6	38.1	24.0
TROD	51.2	45.9	27.4	18.8	16.1	10.7	10.4	41.4	25.0
MPCA	52.4	46.3	29.8	18.0	17.7	11.5	10.8	42.5	26.2
UMPCA	28.9	22.9	11.9	5.7	5.5	3.5	5.4	21.2	11.6
PCA+LDA	**74.8**	54.6	38.6	**23.6**	17.9	**18.8**	**12.7**	**55.5**	**33.8**
ULDA	61.2	45.2	29.3	17.4	13.8	11.5	10.1	45.1	26.5
R-JD-LDA	69.6	53.9	35.2	20.2	16.8	14.6	12.5	51.9	31.0
DATER	61.9	54.4	35.0	17.2	16.1	12.8	11.8	49.6	28.8
GTDA	61.5	54.1	**40.2**	18.3	16.1	10.1	10.1	51.3	29.2
TR1DA	65.3	54.1	36.9	18.0	16.3	12.3	12.0	51.8	30.2
MPCA-S	57.9	52.2	34.5	17.3	**18.2**	12.3	11.6	47.0	27.9
MPCA+LDA	**74.3**	**61.0**	**43.8**	**23.0**	**19.8**	**17.2**	**14.6**	**58.9**	**35.5**
R-UMLDA	52.3	45.6	25.5	7.6	3.9	5.5	3.5	40.8	19.9
R-UMLDA-A	68.9±1.5	**59.3**±0.6	36.3±1.2	14.5±1.0	13.1±1.4	10.0±1.1	9.1±0.6	53.8± 1.1	29.0±0.9

From all the three tables and the figures, MPCA+LDA achieves the best overall performance, demonstrating again the power of this algorithm. The PCA+LDA algorithm also gives good recognition performance, especially in classifying individual gait samples. The overall results of the MPCA-S algorithm is competitive as well. It should be noted that the supervised MPCA-S algorithm results in better recognition rates than the unsupervised MPCA algorithm, especially in the rank 1 identification rates of gait sequences, indicating the effectiveness of the discriminative feature selection scheme. Among the unsupervised algorithms, PCA has the best CRRs for individual samples. For rank 1 and rank 5 identification rates, the performance of MPCA and that of PCA are close

Table 7.8: Gait recognition results on the $32 \times 22 \times 10$ USF gait database V.1.7: the rank 1 identification rate (%) for sequences. MeanABC is the average over probes A, B, and C and MeanAll is the average over all seven probes.

Probe	A	B	C	D	E	F	G	MeanABC	MeanAll
PCA	80.3	82.9	56.1	32.8	25.6	22.4	16.3	73.1	43.9
CSA	69.0	65.9	43.9	25.4	25.6	14.9	16.3	59.6	36.5
TROD	74.6	73.2	53.7	26.9	**27.9**	13.4	14.0	67.2	39.6
MPCA	84.5	80.5	56.1	29.9	25.6	20.9	16.3	72.1	43.5
UMPCA	66.2	56.1	31.7	9.0	7.0	3.0	4.7	51.3	24.8
PCA+LDA	93.0	73.2	58.5	**40.3**	**27.9**	**26.9**	18.6	74.9	**47.4**
ULDA	87.3	58.5	51.2	26.9	18.6	23.9	16.3	62.8	36.5
R-JD-LDA	88.7	70.7	56.1	35.8	**27.9**	25.4	18.6	71.4	43.3
DATER	85.9	78.0	53.7	28.4	20.9	16.4	20.9	71.7	40.2
GTDA	88.7	75.6	65.9	25.4	18.6	13.4	18.6	76.7	43.0
TR1DA	85.9	75.6	61.0	23.9	25.6	17.9	**23.3**	72.3	42.4
MPCA-S	90.1	**85.4**	**68.3**	32.8	20.9	20.9	14.0	**80.0**	45.0
MPCA+LDA	**98.6**	**87.8**	**75.6**	**40.3**	**32.6**	**26.9**	**25.6**	**84.3**	**53.1**
R-UMLDA	87.3	73.2	46.3	9.0	4.7	6.0	2.3	66.3	30.4
R-UMLDA-A	**94.6**±1.6	78.2±2.9	60.2±2.4	18.3±3.4	18.6±2.9	15.0±3.9	16.2±1.4	75.6±1.9	39.5±1.4

and they outperform the other unsupervised algorithms. In particular, the better performance of MPCA over CSA shows again that centering is good for recognition. The UMPCA algorithm has the worst results because it can extract at most ten features, which are not sufficient for good recognition. Similar to UMPCA, R-UMLDA is also restricted in the number of useful features so its results are not good either. Nonetheless, as supervised methods, R-UMLDA and R-UMLDA-A have better results than UMPCA and on the easier probes (A, B, and C), the R-UMLDA-A algorithm is among the top performing algorithms. Finally, it should be pointed out that as in face recognition, the performance of the MPCA+LDA algorithm, as well as PCA+LDA, on gait recognition

Table 7.9: Gait recognition results on the $32 \times 22 \times 10$ USF gait database V.1.7: the rank 5 identification rate (%) for sequences. MeanABC is the average over probes A, B, and C and MeanAll is the average over all seven probes.

Probe	A	B	C	D	E	F	G	MeanABC	MeanAll
PCA	95.8	85.4	78.0	56.7	46.5	49.3	48.8	86.4	65.4
CSA	88.7	82.9	68.3	50.7	44.2	38.8	39.5	78.7	56.7
TROD	91.5	85.4	68.3	52.2	46.5	37.3	41.9	81.7	59.0
MPCA	95.8	85.4	78.0	56.7	48.8	47.8	46.5	86.4	65.4
UMPCA	87.3	75.6	58.5	28.4	25.6	19.4	25.6	73.4	43.5
PCA+LDA	98.6	80.5	75.6	**59.7**	48.8	**55.2**	46.5	84.9	**66.4**
ULDA	90.1	78.0	58.5	49.3	41.9	44.8	34.9	75.6	55.3
R-JD-LDA	95.8	85.4	78.0	53.7	48.8	52.2	39.5	86.4	63.7
DATER	95.8	82.9	73.2	56.7	**53.5**	53.7	48.8	83.1	64.4
GTDA	95.8	82.9	78.0	53.7	44.2	46.3	46.5	84.8	62.5
TR1DA	94.4	85.4	75.6	53.7	46.5	44.8	48.8	83.8	61.6
MPCA-S	**100.0**	**92.7**	**85.4**	55.2	48.8	50.7	**53.5**	**92.7**	65.8
MPCA+LDA	**100.0**	**92.7**	**87.8**	**70.1**	**62.8**	**62.7**	**51.2**	**93.5**	**74.7**
R-UMLDA	98.6	82.9	75.6	34.3	27.9	29.9	18.6	85.7	51.2
R-UMLDA-A	99.9±0.3	82.9±0.0	77.8±1.1	48.4±2.4	43.6±2.7	39.3±1.4	32.3±2.9	86.9±0.3	58.8±1.3

is sensitive to the MPCA/PCA dimensionality before LDA too, which will be illustrated in the next set of experiments.

7.4.2 Comparison with the state-of-the-art gait recognition algorithms

As illustrated in Fig. 1.1 (page 2), a gait recognition system typically consists of the following components: algorithm to partition a gait sequence into cycles (or half cycles), feature representation and extraction method, and the matching algorithm. Different processing steps are expected to have an impact on the recognition results. Since the focus of this dissertation is on feature extraction, simple procedures have been adopted for preprocessing and matching.

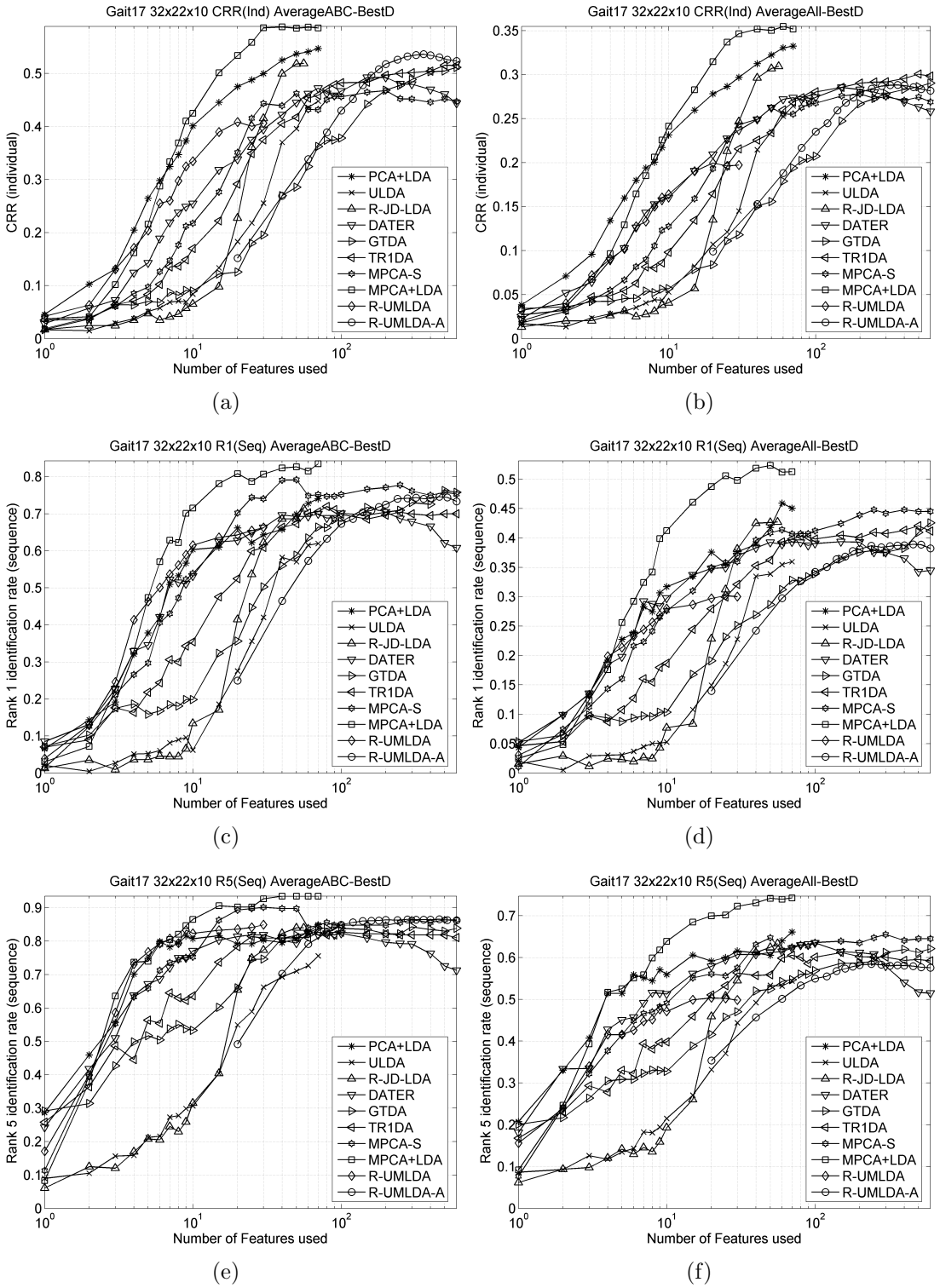


Figure 7.8: Supervised subspace learning results on the $32 \times 22 \times 10$ USF gait database V.1.7. The average over probes A, B, and C: (a) CRR for individual samples, (c) rank 1 identification rate for sequences, and (e) rank 5 identification rate for sequences. The average over all seven probes: (b) CRR for individual samples, (d) rank 1 identification rate for sequences, and (f) rank 5 identification rate for sequences.

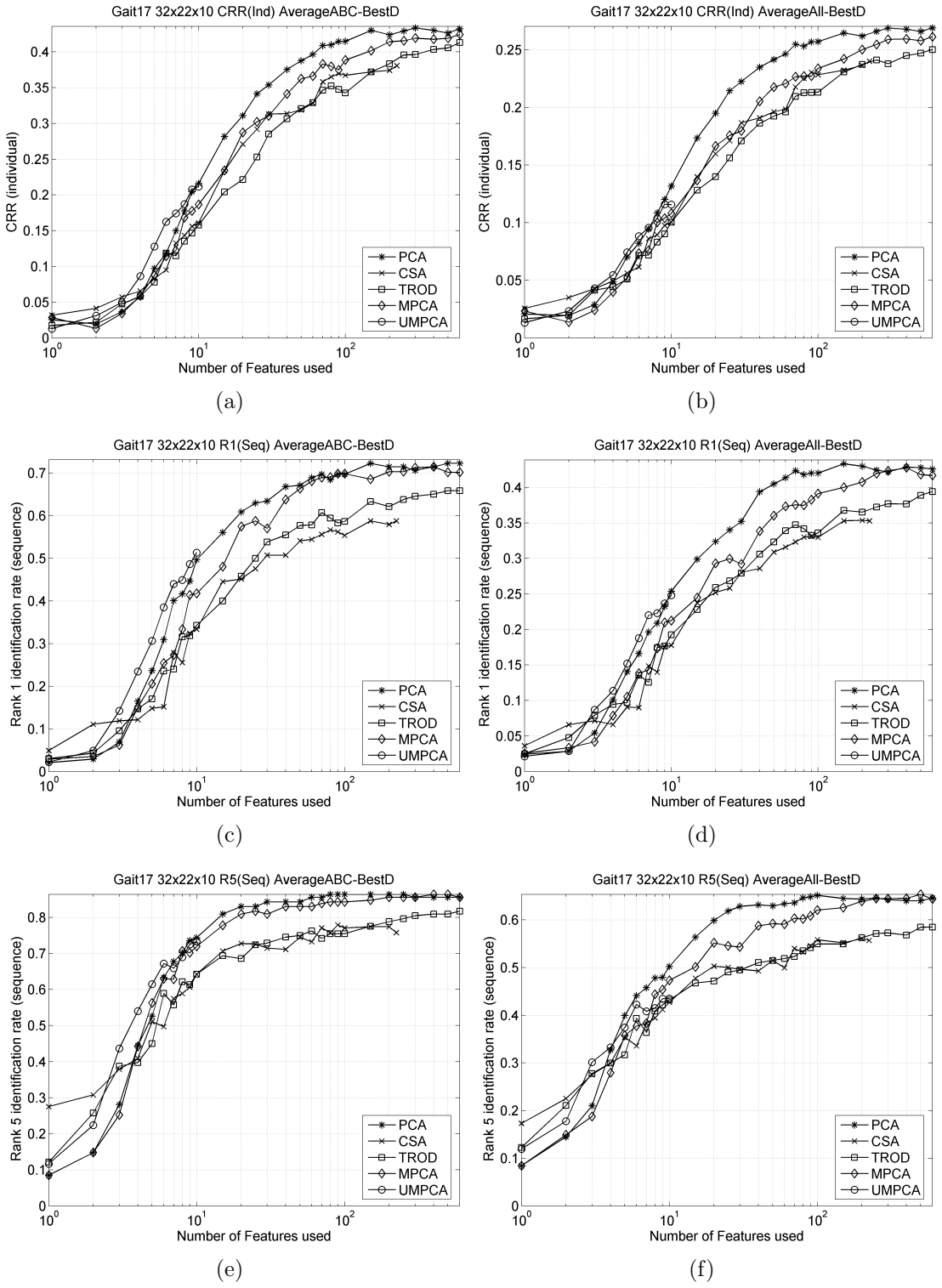


Figure 7.9: Unsupervised subspace learning results on the $32 \times 22 \times 10$ USF gait database V.1.7. The average over probes A, B, and C: (a) CRR for individual samples, (c) rank 1 identification rate for sequences, and (e) rank 5 identification rate for sequences. The average over all seven probes: (b) CRR for individual samples, (d) rank 1 identification rate for sequences, and (f) rank 5 identification rate for sequences.

In this subsection, the MPCA-S and MPCA+LDA algorithms are compared against the state-of-the-art gait recognition algorithms, which employ more sophisticated preprocessing and/or matching algorithms, and the baseline algorithm provided in the "Gait Challenge" problem [109]. The algorithms to be compared, as summarized in Table 7.10 in their original formulations, are the baseline algorithm [109], the Hidden Markov Model (HMM) framework [49] using the entire silhouette as the feature (the direct approach), the linear time normalization (LTN) algorithm [6] using the silhouette feature, and the Gait Energy Image (GEI) algorithm [34]. Table 7.10 illustrates the differences between algorithms. It should be pointed out that in the HMM approach, besides feature extraction and matching, HMM parameter estimation (training) is a major component too and it is not shown in the table.

Table 7.10: Comparison of the state-of-the-art gait recognition algorithms.

Approach	Preprocessing	Cycle partition	Feature extraction	Matching
Baseline [109]	Baseline	Period estimation from median of minima distances	Silhouettes	Spatial-temporal correlation
HMM [49]	Baseline	Adaptive filter	Silhouettes	Viterbi algorithm
LTN [6]	Baseline+silhouette refinement	Autocorrelation, optimal filter, merging	Silhouettes	LTN distance with symmetric matching
GEI [34]	Baseline	Maximum entropy spectrum estimation	PCA+LDA on averaged silhouette	Minimum Euclidean distance to class mean
MPCA-S	Baseline+temporal linear interpolation	Running average filter	MPCA	Nearest neighbor with symmetric matching
MPCA+LDA			MPCA+LDA	

This set of gait recognition experiments is performed on the USF gait database V.1.7 with the original full size of $128 \times 88 \times 20$. For the MPCA-based algorithms (MPCA-S and MPCA+LDA), the seven distance measures in Table 3.1 (page 40) are tested. For MPCA-S, the feature length $H = H_{\mathbf{y}}$ and the weight vector is $\mathbf{w}$, where $H_{\mathbf{y}}$ and $\mathbf{w}$ are defined in Sec. 4.4 (page 78). For MPCA+LDA, $H = H_{\mathbf{z}}$ and $\mathbf{w}(h) = \sqrt{\lambda_h}$, where $H_{\mathbf{z}}$ and λ_h are also defined in Sec. 4.4. For different $H_{\mathbf{y}}$ (up to 800) and for the seven distance measures listed, the average rank 1 and rank 5 identification rates are plotted in

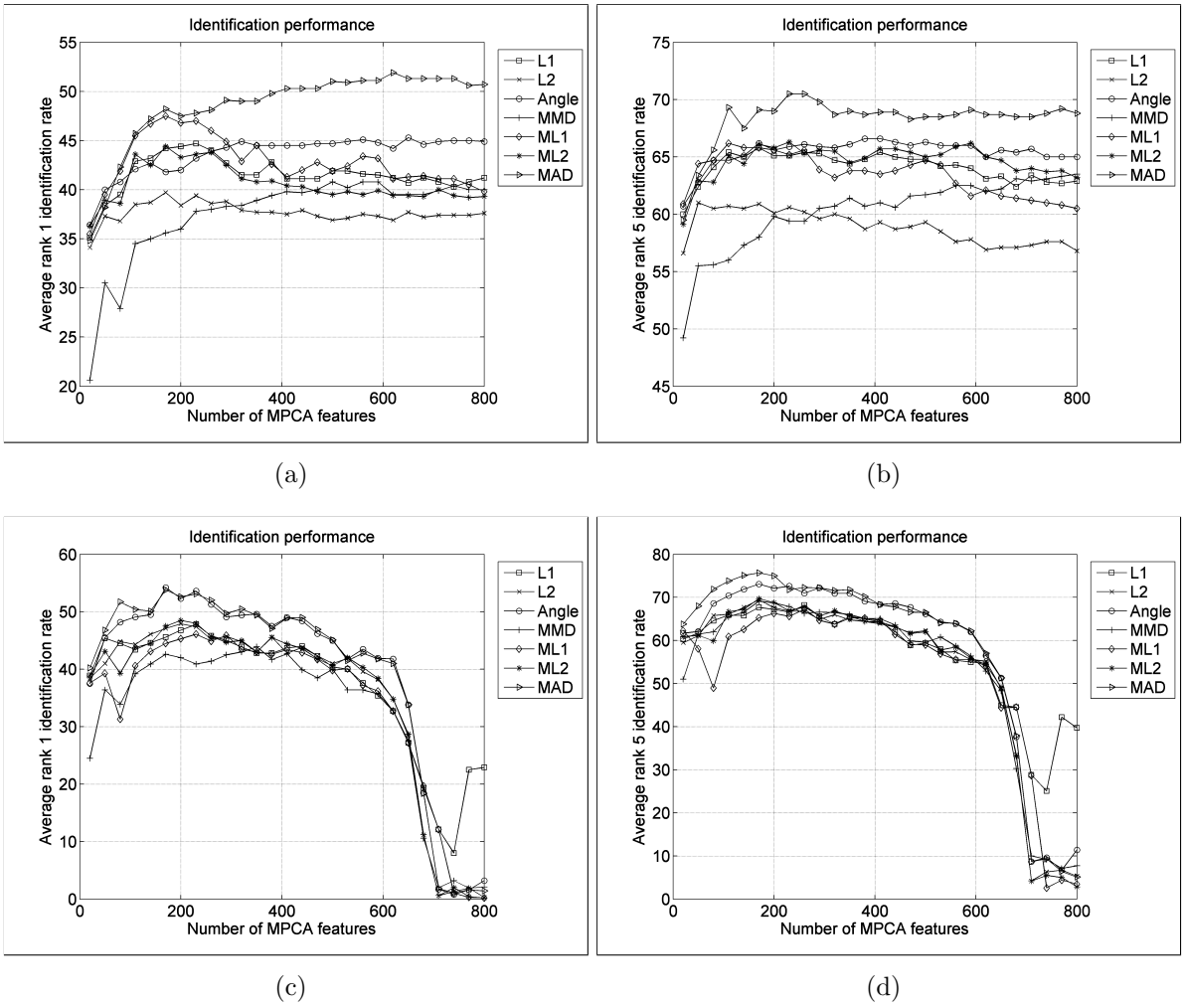
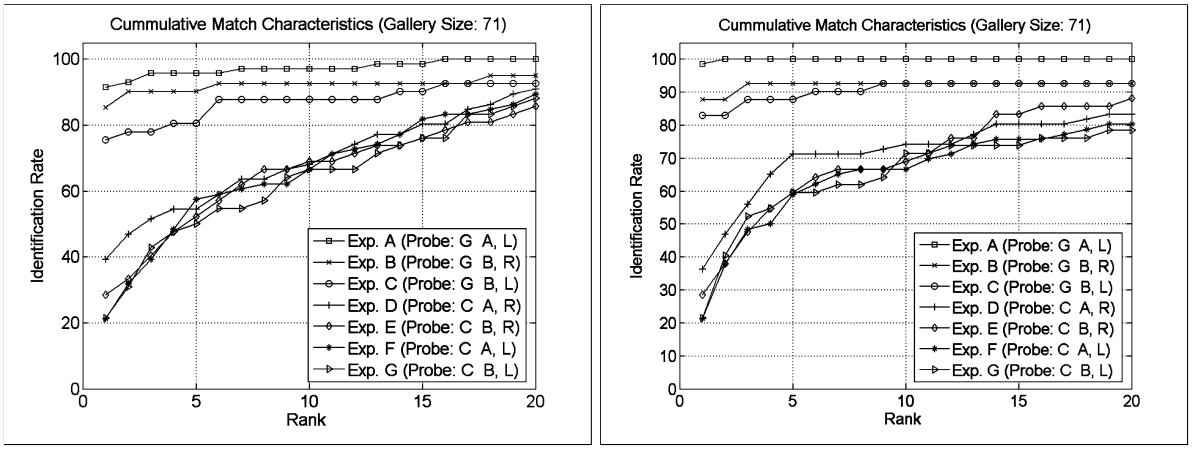


Figure 7.10: Gait recognition results against the number of MPCA features used for the seven distance measures: (a) the rank 1 and (b) rank 5 identification performance of the MPCA-S algorithm; (c) the rank 1 and (d) rank 5 identification performance of the MPCA+LDA algorithm.

Figs. 7.10(a) and 7.10(b), respectively, for the MPCA-S approach, and in Figs. 7.10(c) and 7.10(d), respectively, for the MPCA+LDA approach. For the MPCA-S approach, the MAD measure, with the proposed weight vector $\mathbf{w}$, significantly outperforms all the other distance measures for $H_{\mathbf{y}} > 200$, demonstrating the effectiveness of the proposed weighting scheme. For the MPCA+LDA approach, the angle and MAD measures outperform all the other measures at rank 1 and the MAD measure is better than the angle at rank 5 for $H_{\mathbf{y}} < 200$. Thus, both approaches choose MAD as the distance measure in



(a) (b)

Figure 7.11: The CMC curves of the gait recognition results up to rank 20 for (a) the MPCA-S algorithm, and (b) the MPCA+LDA algorithm.

comparison against the state-of-the-art gait recognition algorithms. With MAD, when more and more MPCA features are included, i.e., $H_{\mathbf{y}}$ increases, the identification rate for MPCA-S keeps increasing first, except some small fluctuations, and becomes steady beyond a certain point, indicating that most of the MPCA features selected first are good features for classification. For the MPCA+LDA approach, due to the LDA feature extraction process where the maximum feature vector length is $C - 1$, the performance against $H_{\mathbf{y}}$ is different: with the MAD measure, the identification rates (rank 1 and 5) reach maximum around $H_{\mathbf{y}} = 200$ and drop at a higher rate for $H_{\mathbf{y}} > 600$, suggesting that more MPCA features (than 200) may lower the performance of MPCA+LDA. Thus, for best performance in gait recognition, MPCA+LDA needs appropriate determination of the dimensionality ($H_{\mathbf{y}}$) before LDA as well.

Based on empirical study, the best gait recognition performance for the MPCA-S and MPCA+LDA approaches is obtained with $H_{\mathbf{y}} = 620$ and $H_{\mathbf{y}} = 170$, respectively, using MAD. The detailed results are depicted using the CMCs in Figs. 7.11(a) and 7.11(b). They are compared with the state-of-the-art gait recognition algorithms in Tables 7.11 and 7.12, where the rank 1 and rank 5 identification rates are listed for each probe (A to G) together with their averages, respectively. From the tables, HMM, LTN, GEI, MPCA-

S, and MPCA+LDA algorithms have no significant difference in the rank 1 performance, although LTN is slightly poorer, and they outperform the baseline results by more than 19%. For the rank 5 performance, MPCA+LDA has the best performance.

Table 7.11: The state-of-the-art gait recognition results on the full size USF gait database V.1.7: the rank 1 identification rate (%) for sequences. MeanAll is the average over all seven probes.

Probe	A	B	C	D	E	F	G	MeanAll
Baseline	79	66	56	29	24	**30**	10	42
HMM	99	**89**	78	35	29	18	24	53
LTN	94	83	78	33	24	17	21	50
GEI	**100**	85	80	30	**33**	21	**29**	**54**
MPCA	92	85	76	**39**	29	21	21	52
MPCA+LDA	99	**88**	**83**	36	29	21	21	**54**
B-LDA-MPCA	**100**	**88**	**85**	**39**	**34**	**26**	**32**	**58**

Table 7.12: The state-of-the-art gait recognition results on the full size USF gait database V.1.7: the rank 5 identification rate (%) for sequences. MeanAll is the average over all seven probes.

Probe	A	B	C	D	E	F	G	MeanAll
Baseline	96	81	76	61	55	46	33	64
HMM	**100**	90	**90**	**65**	**65**	**60**	50	**74**
LTN	99	85	83	**65**	**67**	58	48	72
GEI	**100**	85	88	55	55	41	48	67
MPCA	96	90	81	55	52	58	50	69
MPCA+LDA	**100**	**93**	88	**71**	60	**59**	**60**	**76**
B-LDA-MPCA	**100**	**93**	**90**	63	59	54	**52**	73

From the comparisons in Table 7.10, the performance of the HMM framework is mainly contributed to the adaptive filter used for cycle partition, the Viterbi algorithm used for probabilistic matching, and the iterative training of the HMM parameters, while the performance of LTN is mainly contributed to the silhouette refinement, the robust

cycle partition procedure, and the LTN distance matching strategy. Besides PCA+LDA in feature extraction, the GEI algorithm utilizes a robust estimator as well for the cycle partition taking into account of the periodic nature of the gait signal and it seems that this tends to improve the recognition performance. To summarize, silhouette refinement could be a beneficial step for gait recognition and robust cycle partition seems to be an important component in these state-of-the-art gait recognition algorithms. In addition, robust matching algorithms such as the Viterbi algorithm used in HMM have great potential for gait recognition as well.

On the whole, despite a design without optimizing the preprocessing procedures, cycle partition method, and matching algorithm, the MPCA-based approach to gait recognition achieves highly competitive performance and compares favorably to the state-of-the-art gait recognizers. This indicates that the MPCA-based approach is a very promising tool for gait recognition. Its performance can be further improved by silhouette refinement, and robust cycle partition and matching algorithms.

7.4.3 Gait recognition with MPCA+Boosting

As mentioned earlier, the boosting framework has been shown to be effective in face recognition [93], but no similar study has been done for gait recognition. This section evaluates the effectiveness of the B-LDA-MPCA algorithm in enhancing the gait recognition performance. In particular, the effects of the gait feature vector dimension $H_{\mathbf{y}}$ and the regularization parameter κ are studied, in addition to ξ, the number of LDA training samples per class. This set of experiments uses the full-size USF gait database V.1.7 as well.

In B-LDA-MPCA, MPCA is applied to get the gait feature vectors $\{\mathbf{y}_m\}$ for the input to the booster. As in [93], the output dimension $H_{\mathbf{z}}$ of the LDA learner is fixed at 35, which is not optimized, and the maximum number of iterations is set to $T = 60$. The best performing set of parameters for the B-LDA-MPCA algorithm is $\xi = 3$, $H_{\mathbf{y}} = 180$,

and $\kappa = 10^{-2}$ in the test. The evolutions of various CRRs over the boosting steps are shown in Fig. 7.12(a) with this set of parameters. In the figure legends, 'Gal' means the CRRs for the gallery set and 'Prb' denotes the average CRRs for the seven probe sets. The CRRs for individual gait samples and gait sequences are denoted by 'Ind' and 'Seq', respectively. The CRRs obtained from the single learner in each step are denoted as 'Sgl' and the CRRs obtained from the aggregated learners are denoted by 'Bst'. For instance, 'PrbSeqBst' is the average CRR of all probe sequences obtained from the combined learners. From the figure, the effectiveness of the boosting scheme is observed. The CRRs for the probe samples (sequences) produced by the single learners are around 20% (below 40%), while the CRRs by the boosted learners are around 40% (near 60%), which is a boost of about 20% in the CRR.

The gait recognition results of B-LDA-MPCA on each probe set and their average are shown in Tables 7.11 and 7.12 with the best parameter set above for $T = 24$. In the rank 1 identification rate, B-LDA-MPCA has improved over MPCA+LDA consistently on each probe set except probe B where there is no improvement. On probe F, the improvement of 11% is the greatest, and the improvement in the average CRR is 4%. The consistent improvement shown over both easy probes (A, C) and difficult probes (D, E, F, G) in rank 1 identification rate demonstrates the effectiveness of the proposed solution. However, from Table 7.12, its average rank 5 identification rate is lower than the MPCA+LDA solution with the same parameter settings.

The effects of ξ, $H_{\mathbf{y}}$, and κ on the gait recognition performance of MPCA+boosting are shown in Figs. 7.12(b), 7.12(c), and 7.12(d), respectively. Since it is not possible to show the results of all possible parameter combinations, the effects of a parameter are shown by fixing all the others. The fixed set of parameters is chosen to be the best set above: $\xi = 3$, $H_{\mathbf{y}} = 180$, and $\kappa = 10^{-2}$. In the following, only the results 'PrbSeqBst' will be shown.

The minimum number of samples in a class is 7 for the gallery set. Therefore, ξ

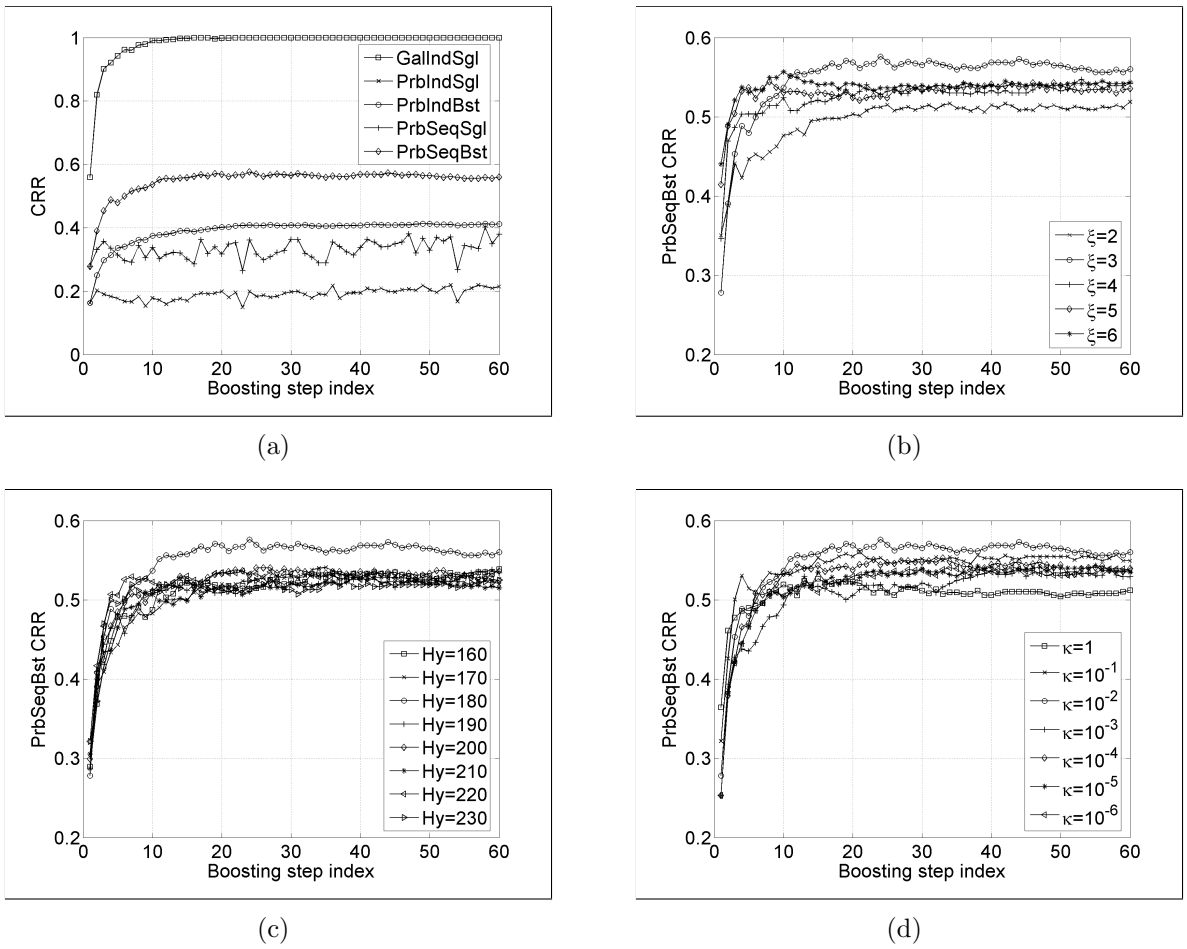


Figure 7.12: Illustrations of MPCA+boosting on gait recognition: (a) the evolutions of various CRRs over the boosting steps with the best parameter set; the effects of (b) ξ, (c) $H_{\mathbf{y}}$, and (d) κ.

ranging from 2 to 6 is tested and the results are shown in Fig. 7.12(b). Similar results as in [93] have been observed here. It shows that the learner cannot be too weak or too strong for the booster to have a positive effect.

B-LDA-MPCA introduces an additional control of learner weakness by $H_{\mathbf{y}}$ and its effects are shown in Fig. 7.12(c). As mentioned in Section 7.4.2, the optimal $H_{\mathbf{y}}$ for a single LDA learner is 200, while the figure shows that the weakened learners with $H_{\mathbf{y}} = 180$ give a much better boosting result than the stronger learners with $H_{\mathbf{y}} = 200$. Hence, $H_{\mathbf{y}}$ affects the booster in a similar way as ξ.

Finally, the effects of the regularization are shown in Fig. 7.12(d). This figure shows

that an appropriate regularization parameter κ does result in better generalization. This study demonstrates that gait recognizer can benefit from making use of the fact that the within-class scatter of gait patterns under various capturing conditions is greater than that under the same capturing condition.

Table 7.13: Summary of the performance and computational complexity of MPCA, UMPCA, MPCA+LDA, and R-UMLDA-A.

Unsupervised task (computational complexity)*	MPCA (low)	UMPCA (medium)
Low-resolution face recognition	No. 2 out of 6	No. 5 out of 6
High-resolution face recognition	No. 2 out of 6	No. 1 out of 6
Gait recognition	No. 2 out of 5	No. 5 out of 5
Supervised task (computational complexity)	MPCA+LDA (low)	R-UMLDA-A (high)
Face recognition with $L = 2$	No. 1 out of 10	No. 4 out of 10
Face recognition with $L = 3$	No. 1 out of 10	No. 5 out of 10
Face recognition with $L = 4$	No. 4 out of 10	No. 1 out of 10
Face recognition with $L = 5, 6$	No. 2 out of 10	No. 1 out of 10
Face recognition with $L = 8 \sim 40$	No. 1 out of 10	No. 2 out of 10
Face recognition with $C = 80$	No. 2 out of 10	No. 1 out of 10
Face recognition with $C = 160, 240, 320$	No. 1 out of 10	No. 2 out of 10
Gait recognition on the same surface	No. 1 out of 10	No. 4 out of 10
Gait recognition on same/different surfaces	No. 1 out of 10	No. 8 out of 10

*The performance is indicated by the ranking among algorithms compared, e.g., "No. 2 out of 10" means that the algorithm is the second best algorithm out of ten algorithms compared. The computational complexity is indicated by a rough ranking in terms of low, medium, and high.

7.5 Discussions on Face and Gait Recognition Results

Extensive experiments have been performed on face and gait recognition to evaluate the proposed algorithms. Table 7.13 gives a summary of the face and gait recognition

performance as well as the computational complexity for the four proposed (two unsupervised and two supervised) solutions: MPCA, MPCA+LDA, UMPCA, and R-UMLDA-A. From the results presented, several important observations have been made and they are summarized below:

1. The MPCA+LDA algorithm has the best overall performance in both face and gait recognition under various scenarios although the MPCA dimensionality before LDA needs to be set properly for the best performance. The advantages of MPCA+LDA over PCA+LDA are due to the proposed MPCA algorithm since the only difference between the two algorithms is MPCA features versus PCA features. Extracting features directly from the tensorial data indeed results in more useful representation on both face and gait data.

2. The MPCA algorithm with discriminative feature selection outperforms the MPCA operating in the unsupervised mode, showing the effectiveness of the proposed feature selection procedure. The advantage here comes from taking class-specific information into account.

3. The combination of MPCA and boosting also have obtained improved rank 1 identification results over MPCA+LDA in gait recognition, demonstrating possible benefits of the integrated solution. The improvement here attributes to the good generalization capability of the boosting framework.

4. It is also worth noting the performance difference between CSA and MPCA. The only algorithmic difference is the centering of the input data before feature extraction. The experimental results on face and gait recognition show that in most cases, MPCA outperforms CSA. This indicates that centering is indeed beneficial for recognition tasks since the true variation of the data with respect to the data center rather than the origin is captured.

5. The results of UMPCA are not competitive in challenging experimental conditions due to two reasons. The first reason is that it is an unsupervised algorithm without considering the class-specific information. The second reason is that the maximum number of uncorrelated features extracted by UMPCA is limited (see Corollary 5.1 on page 103). However, in the setting of unsupervised learning on a higher-resolution data set, it outperforms other unsupervised learning algorithms, particularly in the low-dimensional subspace, despite the fact that it captures much lower variation. In this case, the uncorrelated features directly extracted from tensorial data are shown to be more useful. In addition, as an unsupervised method, it can also handle the difficult one training sample scenario.

6. Although random initialization is involved in R-UMLDA-A, from the standard deviations based on 20 repeated trials reported in Tables 7.3, 7.5, 7.7, 7.8, and 7.9, the recognition results obtained by R-UMLDA-A have low variance. This demonstrates that R-UMLDA-A is a stable algorithm despite of the random initialization involved.

7. The proposed R-UMLDA-A algorithm, without tuning the regularization parameter, has achieved good overall performance in face recognition experiments under various scenarios. Its performance for gait recognition on the same surface is also competitive against other algorithms. Its advantage over MPCA+LDA is that there is no need to search for the best performing set of parameters and its best performance is always achieved around the same parameter setting, as evident in Figs. 7.2, 7.4, and 7.8. This implies that R-UMLDA-A is a robust and effective recognition algorithm for tensor objects, attributing to uncorrelated discriminative feature extraction directly from tensorial data, and the regularization and aggregation schemes proposed. It will be particularly useful when there is no sufficient validation data available for searching the best parameter settings.

Based on the results and the discussions above, the practical recommendations for the choice of the proposed algorithms are outlined here. Since among all the algorithms compared, the MPCA+LDA algorithm has achieved the best overall performance on both face and gait recognition problems, it should be chosen as long as a good MPCA dimensionality before LDA can be determined. However, when this is not possible, R-UMLDA-A is a safe choice since the performance of MPCA+LDA may be dependent on the MPCA dimensionality before LDA while in contrast, R-UMLDA-A is less sensitive to parameter settings. UMPCA is more suitable for unsupervised learning tasks that requires only a small number of features and for tensor objects of high resolution. In terms of computational cost, MPCA+LDA is a highly efficient algorithm due to the good convergence performance of MPCA while R-UMLDA-A is more computationally expensive in the training process since a number of UMLDA feature extractors need to be trained. The computational complexity of UMPCA is similar to a single UMLDA. Finally, the recommended parameter settings for these three proposed solutions are listed in Table 7.14. Nevertheless, when a validation data set is available, it is always preferred to tune the subspace dimensionality and other parameters of the chosen algorithm for its best performance.

Table 7.14: Recommended parameter settings for MPCA+LDA, UMPCA, and R-UMLDA-A.

Setting	MPCA+LDA	UMPCA	R-UMLDA-A
Initialization	full projection truncation Sec. 4.3.2 (page 70)	uniform initialization (normalized $\mathbf{1}$)	uniform initialization for $a = 1, 5, 9, 13, 17$, random initialization for other values of a
Projection order	from 1 to N	from 1 to N	from 1 to N
Maximum iteration K	1	10	10
Subspace dimensionality	$C - 1$	$\min\{\min_n I_n, M\}$	$A = 20$, $P = 15 \sim 20$
Other parameters	$H_{\mathbf{y}}$ determined by a validation set, $Q = 97$		γ_a sampled uniformly in log scale from the interval $[10^{-7}, 10^{-2}]$

7.6 Summary

This chapter presents experimental results on face and gait recognition problems in order to evaluate the algorithms proposed in this research against existing linear and multilinear subspace learning algorithms.

Three sets of face recognition experiments have been carried out on the PIE and FERET databases. The PIE database is used to evaluate all the algorithms under varying number of training samples per class. MPCA+LDA and R-UMLDA-A algorithms outperform the other algorithms in most cases. A FERET database is then constructed to study the performance of supervised subspace learning algorithms against different number of classes, where MPCA+LDA and R-UMLDA-A give the best results. The third set of experiments evaluate the unsupervised subspace learning algorithms on another higher-resolution, and less challenging FERET database. UMPCA has been shown to outperform the other unsupervised learning algorithms significantly, especially in the low-dimensional space. Moreover, the one training sample scenario can be handled by UMPCA as well.

The USF gait database is used for three sets of gait recognition experiments. All the algorithms are first evaluated on the downsampled low-resolution gait database, with the MPCA+LDA solution performing the best. Next, the MPCA-based approach is compared with the state-of-the-art gait recognition algorithms with more sophisticated preprocessing and matching algorithms. The results indicate that the MPCA-based approach is a very promising tool for gait recognition and its performance can be further improved by silhouette refinement, robust cycle partition, and advanced matching algorithms. Lastly, the integration of the boosting technology and MPCA is studied on the gait recognition problem with improved rank 1 recognition rate observed.

In summary, the experimental evaluations demonstrate that MPCA+LDA gives the best overall performance but it needs the determination of the appropriate MPCA dimensionality before LDA. In contrast, the R-UMLDA-A algorithm offers competitive

performance in most cases and it has good stability with the same parameter setting. In unsupervised learning scenarios, especially in low-dimensional subspace, UMPCA has been shown to have good performance. With both algorithms and experimental results presented, the next chapter draws the conclusions of this dissertation by recapitulating the major contributions and pointing out future directions of research.

Chapter 8

Conclusions

This dissertation has focused on investigating the multilinear subspace learning approach for appearance-based face and gait recognition. It has contributed to both understanding and developing of multilinear subspace learning algorithms. In this chapter, the key contributions of the dissertation are summarized and directions for future research in related areas are provided.

8.1 Key Contributions

The problems of appearance-based face and gait recognition are challenging due to the large variability of the appearance, the high complexity of pattern distribution, and the insufficiency of training samples. Subspace learning, or dimensionality reduction, attempts to project high-dimensional data to a low-dimensional space where the recognition task is easier, and it has become the arguably most successful approach in appearance-based learning. However, traditional subspace learning algorithms, such as PCA and LDA, are linear methods. They have to reshape face or gait objects, which have natural tensorial representations, into very-high-dimensional vectors before learning. This reshaping leads to high computational and memory demand and the need to estimate a very large number of parameters. It also breaks the natural structure and correlation in

the original multidimensional representation. In contrast, multilinear subspace learning treats tensorial objects in their original form and has the potential to learn more compact and useful representations for better recognition.

This dissertation has contributed in two ways. The first contribution is a framework that unifies multilinear subspace learning and explains both existing multilinear subspace learning algorithms and those proposed in this dissertation. It has also contributed through the development of a number of multilinear subspace learning algorithms. For completeness, these contributions are summarized below.

1. Multilinear subspace learning relies on multilinear projection, but as a new technology, a systematic treatment on this topic was not available in the literature. This research started by addressing this topic first. The basics of multilinear projection have been thoroughly covered. Three basic types of multilinear projections have been categorized as: the vector-to-vector projection, the tensor-to-tensor projection, and the tensor-to-vector projection. The connections between these three projections have also been analyzed in depth. In addition, several tensor-based and scalar-based scatter measures have been defined to assist the understanding and development of multilinear subspace learning algorithms. Under the framework of the multilinear subspace learning introduced here, existing multilinear subspace learning algorithms have been understood better and new algorithms have been developed.

2. The MPCA algorithm has been proposed for analysis of tensor objects. It is a multilinear extension of PCA. MPCA determines a tensor-to-tensor projection that captures most of the signal variation present in the original tensorial representation. Issues due to the iterative nature of the algorithm, including initialization, projection order, termination, convergence, and subspace dimensionality determination, have been addressed in detail. A discriminative MPCA feature selection

procedure has been further proposed and the MPCA+LDA algorithm has been formulated. Moreover, the combination of MPCA with the boosting technology has also been investigated. The applications of these MPCA-based feature extraction algorithms on the problems of face and gait recognition have demonstrated their effectiveness in handling various challenging tasks. In particular, the MPCA+LDA algorithm has achieved the best recognition performance in most cases, although the appropriate MPCA feature dimensionality before LDA needs to be determined for the best performance. In addition, the combination of MPCA and the LDA-style booster in [93] has shown the effectiveness of boosting on gait recognition for the first time in the literature.

3. Being aware that PCA derives uncorrelated features, a novel UMPCA algorithm has been further proposed. This algorithm extracts uncorrelated features directly from tensorial data through a tensor-to-vector projection. However, the number of uncorrelated features that can be extracted by UMPCA is no greater than the lowest dimension so it is more suitable for tensor objects with higher resolution or for recognition tasks that need only a small number of features. On the problem of unsupervised face recognition, UMPCA is shown to be particularly effective in the low-dimensional subspace.

4. LDA produces uncorrelated features as well. Thus, in a similar manner as UMPCA, a novel UMLDA algorithm has been proposed, with a regularization mechanism incorporated to address the small sample size problem. The algorithm is affected by initialization and regularization. This observation has led to the introduction of an aggregation scheme to utilize complementary information from differently initialized and regularized UMLDA feature extractors. As UMPCA, UMLDA is also limited in the number of uncorrelated features that can be extracted. Nonetheless, the proposed aggregation scheme has greatly reduced this limitation and also alle-

viated the regularization parameter selection problem. The simulation studies on face and gait recognition problems have shown that the R-UMLDA-A algorithm is effective in face recognition while on gait recognition, it is less competitive, especially under different surfaces. Nevertheless, R-UMLDA-A has been shown to have good stability and its best performance is always achieved around the same parameter setting.

To conclude, from the recognition results and the analysis presented in this dissertation, the three proposed solutions can be ranked in order of algorithmic significance (from high to low) as: MPCA, UMLDA, and UMPCA.

8.2 Future Directions

While many fundamental problems in multilinear subspace learning have been addressed in this dissertation, this is a new field still with many open problems to be considered. This section outlines several research topics that worth further investigation. Two main directions have been identified. One is towards the development of multilinear subspace learning solutions, while the other is towards novel applications where the proposed methods can be applied.

8.2.1 Further development of multilinear subspace learning algorithms

In future research, the algorithms proposed in this dissertation can be further enhanced and new algorithms can be investigated along the following directions:

1. The systematic treatment on multilinear subspace learning in Chapter 2 will benefit the development of new multilinear learning algorithms, especially by extending the rich ideas and algorithms in the linear counterparts to the multilinear case.

This dissertation has focused on the extensions of PCA and LDA to their multilinear counterparts, and the proposed algorithms project input data to a subspace through simple multilinear mapping. In future work, more complicated (nonlinear) mapping can be achieved by developing multilinear extensions of graph-embedding algorithms such as Isomap [125], Locally Linear Embedding [106], and Locality Preserving Projections [38,11]. As mentioned in Section 3.4.5 (page 58), there have been some developments in this area [37,20,151,144,41]. The multilinear subspace learning framework proposed in this dissertation can help the understanding of these existing solutions and it can also benefit further development of multilinear graph-embedding algorithms.

2. The combination of MPCA with LDA and boosting proposed in Chapter 4 has shown promising results on face and gait recognition. Thus, it will be worthwhile to investigate the combination of MPCA with other algorithms, such as neural networks [22] or kernel methods [137,97,89,114] where the input data is mapped to an even-higher dimensional space for better separation. Furthermore, it will also be interesting to study whether the combination of the multilinear algorithms developed here and in the literature, e.g., MPCA with R-UMLDA-A, DATER or GTDA, can lead to more advanced learning algorithms.

3. As pointed out in Chapter 5, a limitation of UMPCA is the limited number of uncorrelated features that can be extracted. In future research, solutions can be sought to extract more features through gradual relaxation of either the zero-correlation constraint or the tensor-to-vector projection constraint. Another direction is to investigate whether the aggregation used in enhancing UMLDA can be useful for UMPCA since UMPCA is sensitive to initialization as well. This dissertation has studied only the aggregation of UMLDA because it is more promising in solving challenging face and gait recognition problems than UMPCA.

4. In the R-UMLDA-A algorithm proposed in Chapter 6, simple regularization and aggregation methods have been adopted. In future work, better regularization mechanisms (e.g., the incorporation of domain knowledge) and other aggregation schemes (such as other combination rules, feature-level fusion, boosting or other ensemble-based learning solutions) can be investigated for possible improvement.

5. This dissertation has examined only linear solutions using vectorial representation and multilinear solutions using natural tensorial representation. It could be an interesting topic to study the hybrid approach for higher-order tensors where a selected number of modes are vectorized to result in tensors with order greater than one but less than N, from which features are extracted.

6. In the proposed algorithms, there are a number of parameters need to be set and they may affect the performance if they are not set properly. For example, the MPCA+LDA algorithm has been shown to be sensitive to the MPCA feature dimensionality $H_{\mathbf{y}}$. In B-LDA-MPCA, the performance depends on the number of boosting steps T, the regularization κ, the MCPA dimensionality $H_{\mathbf{y}}$, and the number of training samples for the LDA learner ξ. In R-UMLDA, the regularization parameter γ can affect the performance too. New ways can be investigated to determine or at least guide the optimal parameter setting automatically or semi-automatically.

7. Finally, in multilinear subspace learning, there are still many unsolved problems remaining, such as the optimal initialization, the optimal projection order, and the optimal stopping criterion. This dissertation has made some attempts in solving some of these problems in MPCA. However, it will be beneficial if further research can lead to deeper understanding on these issues.

8.2.2 Exploring other applications of multilinear subspace learning algorithms

In addition to the tensorial face and gait objects studied in this dissertation, many other real-world data inputs are naturally tensor objects too, as mentioned in Section 1.4.1 (page 6). Consequently, there are a wide range of applications dealing with these real-world tensor objects. In face recognition, besides the traditional 2-D image based approach, high-resolution and three-dimensional face detection and recognition have also emerged as important research directions [7, 66, 14, 101, 69]. Beyond biometric signal analysis, many other computer vision and pattern recognition tasks deal with tensor objects too. Such tasks include image or 3-D object recognition tasks [107] in computer vision, medical image analysis, clustering [148], and content-based retrieval [36], space-time analysis of video sequences for gesture recognition [100] and activity recognition [32] in human-computer interaction (HCI), and space-time super-resolution [117] for digital cameras with limited spatial and temporal resolution.

In addition, many streaming data and mining data are frequently organized as third-order tensors [24, 121, 122]. Data in environmental sensor monitoring are often organized in three modes of time, location, and type [24]. Data in social network analysis are usually organized in three modes of time, author, and keywords [24]. Data in network forensics are often organized in three modes of time, source, and destination, and data in web graph mining are commonly organized in three modes of source, destination, and text [121].

The tensor-object-based applications discussed above are becoming increasingly popular with the advancement in computational power. Thus, it will be interesting to investigate the application of the proposed multilinear subspace learning algorithms in solving these problems. For instance, classical applications of unsupervised learning algorithms, such as unsupervised image categorization, classification, or clustering [148],

can be explored for the MPCA and UMPCA algorithms.

Appendix A

Mathematical Derivations

This appendix provides the mathematical derivations of the five theorems, one lemma, and one corollary presented in this dissertation.

A.1 Proof of Theorem 4.1 in Chapter 4

Proof. The mean tensor of all the projected samples is

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

Write the objective function (4.2) in terms of the input tensor samples as:

$$\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \parallel \mathcal{Y}_m - \bar{\mathcal{Y}} \parallel_F^2 = \sum_{m=1}^{M} \parallel (\mathcal{X}_m - \bar{\mathcal{X}}) \times_1 \tilde{\mathbf{U}}^{(1)^T} \times_2 \tilde{\mathbf{U}}^{(2)^T} ... \times_N \tilde{\mathbf{U}}^{(N)^T} \parallel_F^2 . \tag{A.2}$$

From the definition of the Frobenius norm for a tensor and that for a matrix,

$$\parallel \mathcal{A} \parallel_F = \parallel \mathbf{A}_{(n)} \parallel_F . \tag{A.3}$$

From (2.9), $\Psi_{\mathcal{Y}}$ can be expressed using the equivalent matrix representation by n-mode

unfolding as follows:

$$\Psi_{\mathcal{Y}} = \sum_{m=1}^{M} \parallel \mathbf{Y}_{m(n)} - \bar{\mathbf{Y}}_{(n)} \parallel_F^2 = \sum_{m=1}^{M} \parallel \tilde{\mathbf{U}}^{(n)^T} \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \parallel_F^2, \tag{A.4}$$

Since $\parallel \mathbf{A} \parallel_F^2 = trace(\mathbf{A}\mathbf{A}^T)$, $\Psi_{\mathcal{Y}}$ can be written in terms of the n-mode total scatter matrix of the projected tensor samples:

$$\begin{aligned} \Psi_{\mathcal{Y}} &= \sum_{m=1}^{M} trace\left(\tilde{\mathbf{U}}^{(n)^T} \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right) \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T \cdot \left(\mathbf{X}_{m(n)} - \bar{\mathbf{X}}_{(n)}\right)^T \cdot \tilde{\mathbf{U}}^{(n)}\right) \\ &= trace\left(\tilde{\mathbf{U}}^{(n)^T} \cdot \mathbf{\Phi}^{(n)} \cdot \tilde{\mathbf{U}}^{(n)}\right), \end{aligned} \tag{A.5}$$

Therefore, for given $\tilde{\mathbf{U}}^{(1)}$, ..., $\tilde{\mathbf{U}}^{(n-1)}$, $\tilde{\mathbf{U}}^{(n+1)}$, ..., $\tilde{\mathbf{U}}^{(N)}$, $\Psi_{\mathcal{Y}}$ is maximized if and only if $trace\left(\tilde{\mathbf{U}}^{(n)^T} \cdot \mathbf{\Phi}^{(n)} \cdot \tilde{\mathbf{U}}^{(n)}\right)$ is maximized. The maximum is obtained if $\tilde{\mathbf{U}}^{(n)}$ consists of the P_n eigenvectors of the matrix $\mathbf{\Phi}^{(n)}$ corresponding to the largest P_n eigenvalues. □

A.2 Proof of Lemma 4.1 in Chapter 4

Proof. By successive application of the transpose property of the Kronecker product $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$ [96]:

$$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T = \left(\tilde{\mathbf{U}}^{(n+1)^T} \otimes \tilde{\mathbf{U}}^{(n+2)^T} \otimes ... \otimes \tilde{\mathbf{U}}^{(N)^T} \otimes \tilde{\mathbf{U}}^{(1)^T} \otimes \tilde{\mathbf{U}}^{(2)^T} \otimes ... \tilde{\mathbf{U}}^{(n-1)^T}\right). \tag{A.6}$$

By the Kronecker product theorem $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC} \otimes \mathbf{BD})$ [96],

$$\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T = \left(\tilde{\mathbf{U}}^{(n+1)}\tilde{\mathbf{U}}^{(n+1)^T} \otimes ... \otimes \tilde{\mathbf{U}}^{(N)}\tilde{\mathbf{U}}^{(N)^T} \otimes \tilde{\mathbf{U}}^{(1)}\tilde{\mathbf{U}}^{(1)^T} \otimes ... \tilde{\mathbf{U}}^{(n-1)}\tilde{\mathbf{U}}^{(n-1)^T}\right). \tag{A.7}$$

For all n, when $P_n = I_n$, $\tilde{\mathbf{U}}^{(n)}$ is a square matrix and $\tilde{\mathbf{U}}^{(n)^T}\tilde{\mathbf{U}}^{(n)} = \mathbf{I}_{I_n}$, where $\mathbf{I}_{I_n}$ is an $I_n \times I_n$ identity matrix. Then, $\tilde{\mathbf{U}}^{(n)^{-1}} = \tilde{\mathbf{U}}^{(n)^T}$ and $\tilde{\mathbf{U}}^{(n)}\tilde{\mathbf{U}}^{(n)^T} = \mathbf{I}_{I_n}$. Thus, $\tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}} \cdot \tilde{\mathbf{U}}_{\mathbf{\Phi}^{(n)}}^T = \mathbf{I}_{I_1 \times I_2 \times ... \times I_{n-1} \times I_{n+1} \times ... \times I_N}$. □

A.3 Proof of Theorem 4.2 in Chapter 4

The following lemma explains the relationship between the eigenvalues of two covariance matrices that are closely related.

Lemma A.1. *Let* $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ *be a sample matrix with* I_2 *samples.* $\tilde{\mathbf{X}} \in \mathbb{R}^{I_1 \times P_2}$ *contains* $P_2 < I_2$ *samples from* $\mathbf{X}$*, and* $\mathbf{E} \in \mathbb{R}^{I_1 \times (I_2 - P_2)}$ *contains the rest. Let* λ_{i_1} *and* $\tilde{\lambda}_{i_1}$*, denote the* i_1*th eigenvalues of* $\mathbf{X}\mathbf{X}^T$ *and* $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$*, respectively. Then,* $\tilde{\lambda}_{i_1} \leq \lambda_{i_1}, \text{for } i_1 = 1, ..., I_1$.

Proof. Without loss of generality, let $\tilde{\mathbf{X}} = \mathbf{X}(:, 1 : P_2)$ and $\mathbf{E} = \mathbf{X}(:, (P_2 + 1) : I_2)$. $\mathbf{X}\mathbf{X}^T$ and $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ are both sample covariance matrices and hence symmetric. $\mathbf{E}$ is related to them by $\mathbf{X}\mathbf{X}^T = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T + \mathbf{E}\mathbf{E}^T$. $\mathbf{E}\mathbf{E}^T$ is a covariance matrix as well. Hence, it is positive semidefinite. From the Weyl's theorem [40], which is derived from the Courant-Fisher "min-max theorem" [40], the eigenvalues of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ are not greater than the corresponding eigenvalues of $\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T + \mathbf{E}\mathbf{E}^T = \mathbf{X}\mathbf{X}^T$, i.e., $\tilde{\lambda}_{i_1} \leq \lambda_{i_1}, \text{for } i_1 = 1, ..., I_1$. □

The proof of Theorem 4.2 follows:

Proof. For $n = 1$, $\check{\mathbf{X}}_{m(1)} = \check{\mathbf{U}}^{(1)} \cdot \check{\mathbf{Y}}_{m(1)} \cdot \mathbf{U}^T_{\mathbf{\Phi}^{(1)*}}, m = 1, ..., M$. Thus,

$$\begin{aligned}
\check{\mathbf{\Phi}}^{(1)} &= \sum_m \check{\mathbf{X}}_{m(1)} \mathbf{U}_{\mathbf{\Phi}^{(1)*}} \mathbf{U}^T_{\mathbf{\Phi}^{(1)*}} \check{\mathbf{X}}^T_{m(1)} \\
&= \sum_m \check{\mathbf{U}}^{(1)} \cdot \check{\mathbf{Y}}_{m(1)} \cdot \mathbf{U}^T_{\mathbf{\Phi}^{(1)*}} \mathbf{U}_{\mathbf{\Phi}^{(1)*}} \cdot \mathbf{U}^T_{\mathbf{\Phi}^{(1)*}} \mathbf{U}_{\mathbf{\Phi}^{(1)*}} \cdot \check{\mathbf{Y}}^T_{m(1)} \cdot \check{\mathbf{U}}^{(1)^T} \\
&= \sum_m \check{\mathbf{U}}^{(1)} \cdot \check{\mathbf{Y}}_{m(1)} \check{\mathbf{Y}}^T_{m(1)} \cdot \check{\mathbf{U}}^{(1)^T} = \check{\mathbf{U}}^{(1)} \cdot \sum_m \left(\check{\mathbf{Y}}_{m(1)} \check{\mathbf{Y}}^T_{m(1)} \right) \cdot \check{\mathbf{U}}^{(1)^T},
\end{aligned}$$

where $\mathbf{U}^T_{\mathbf{\Phi}^{(1)*}} \mathbf{U}_{\mathbf{\Phi}^{(1)*}}$ results in an identity matrix. Since $\check{\mathbf{Y}}_{m(1)}$ is simply the first P_1 rows of $\tilde{\mathbf{Y}}_{m(1)}$, $\hat{\lambda}^{(1)}_{i_1} = \lambda^{(1)*}_{i_1}$ for $i_1 = 1, ..., P_1$, and $\hat{\lambda}^{(1)}_{i_1} = 0$ for $i_1 = P_1 + 1, ..., I_1$.

Similarly for $n \neq 1$, $\check{\mathbf{\Phi}}^{(n)} = \check{\mathbf{U}}^{(n)} \cdot \sum_m \left(\check{\mathbf{Y}}_{m(n)} \check{\mathbf{Y}}^T_{m(n)} \right) \cdot \check{\mathbf{U}}^{(n)^T}$. The columns of $\check{\mathbf{Y}}_{m(n)}$ are a subset of the columns of $\tilde{\mathbf{Y}}_{m(n)}$. Therefore, by Lemma A.1, $\hat{\lambda}^{(n)}_{i_n} \leq \lambda^{(n)*}_{i_n}$. Since $\sum_{i_1} \hat{\lambda}^{(1)}_{i_1} < \sum_{i_1} \lambda^{(1)*}_{i_1}$, $\sum_{i_n} \hat{\lambda}^{(n)}_{i_n} = \sum_{i_1} \hat{\lambda}^{(1)}_{i_1} < \sum_{i_n} \lambda^{(n)*}_{i_n} = \sum_{i_1} \lambda^{(1)*}_{i_1}$. Thus, for each mode, at least for one value of i_n, $\hat{\lambda}^{(n)}_{i_n} < \lambda^{(n)*}_{i_n}$. □

A.4 Proof of Theorem 4.3 in Chapter 4

Proof. For the lower bound, considering the 1-mode eigenvalues $\lambda_{i_1}^{(1)*}$ first,

$$\begin{aligned}
\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} &= \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) - \sum_{i_1=1}^{P_1}\sum_{i_2=1}^{P_2}...\sum_{i_N=1}^{P_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) \\
&\geq \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) - \sum_{i_1=1}^{P_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) \\
&= \sum_{i_1=1}^{I_1}\lambda_{i_1}^{(1)*} - \sum_{i_1=1}^{P_1}\lambda_{i_1}^{(1)*} = \sum_{i_1=P_1+1}^{I_1}\lambda_{i_1}^{(1)*}, \qquad \text{(A.8)}
\end{aligned}$$

where $\mathcal{Y}_{var}^*$ is the total scatter tensor (corresponding to the full projection) defined in (4.9). The above inequality can be similarly derived for the other n-mode eigenvalues $\lambda_{i_n}^{(n)*}$, $\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} \geq \sum_{i_n=P_n+1}^{I_n}\lambda_{i_n}^{(n)*}$ for $n = 2, ..., N$. Therefore,

$$\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} \geq \max_n \sum_{i_n=P_n+1}^{I_n}\lambda_{i_n}^{(n)*}. \qquad \text{(A.9)}$$

For the upper bound,

$$\begin{aligned}
\Psi_{\mathcal{X}} - \Psi_{\mathcal{Y}_0} &= \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) - \sum_{i_1=1}^{P_1}\sum_{i_2=1}^{P_2}...\sum_{i_N=1}^{P_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) \\
&\leq \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) - \sum_{i_1=1}^{P_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) \\
&\quad + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) - \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{P_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) \\
&\quad + ... \\
&\quad + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{I_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) - \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}...\sum_{i_N=1}^{P_N}\mathcal{Y}_{var}^*(i_1,i_2,...,i_N) \\
&= \sum_{n=1}^{N}\sum_{i_n=P_n+1}^{I_n}\lambda_{i_n}^{(n)*}. \qquad \text{(A.10)}
\end{aligned}$$

□

A.5 Proof of Theorem 5.1 in Chapter 5

Proof. First, Lagrange multipliers can be used to transform the problem (5.10) to the following to include all the constraints:

$$F(\mathbf{u}_p^{(n^*)}) = \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - \nu\left(\mathbf{u}_p^{(n^*)^T} \mathbf{u}_p^{(n^*)} - 1\right) - \sum_{q=1}^{p-1} \mu_q \mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q, \tag{A.11}$$

where ν and $\{\mu_q, q = 1, ..., p-1\}$ are Lagrange multipliers.

The optimization is performed by setting the partial derivative of $F(\mathbf{u}_p^{(n^*)})$ with respect to $\mathbf{u}_p^{(n^*)}$ to zero:

$$\frac{\partial F(\mathbf{u}_p^{(n^*)})}{\partial \mathbf{u}_p^{(n^*)}} = 2\tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - 2\nu \mathbf{u}_p^{(n^*)} - \sum_{q=1}^{p-1} \mu_q \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = 0. \tag{A.12}$$

Multiplying (A.24) by $\mathbf{u}_p^{(n^*)^T}$ results in

$$2\mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - 2\nu \mathbf{u}_p^{(n^*)^T} \mathbf{u}_p^{(n^*)} = 0 \Rightarrow \nu = \frac{\mathbf{u}_p^{(n^*)^T} \tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u}_p^{(n^*)}}{\mathbf{u}_p^{(n^*)^T} \mathbf{u}_p^{(n^*)}}, \tag{A.13}$$

which indicates that ν is exactly the criterion to be maximized, with the constraint on the norm of the projection vector incorporated.

Next, a set of $(p-1)$ equations are obtained by multiplying (A.24) by $\mathbf{g}_q^T \tilde{\mathbf{Y}}_p^{(n^*)^T}$, $q = 1, ..., p-1$, respectively:

$$2\mathbf{g}_q^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{T_p}^{(n^*)} \mathbf{u}_p^{(n^*)} \; - \; \sum_{q=1}^{p-1} \mu_q \mathbf{g}_q^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \cdot \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = 0. \tag{A.14}$$

Let

$$\boldsymbol{\mu}_{p-1} = [\mu_1 \; \mu_2 \; ... \; \mu_{p-1}]^T \tag{A.15}$$

and use (5.14) and (6.16), then the $(p-1)$ equations of (A.26) can be represented in a

single matrix equation as following:

$$2\mathbf{G}_{p-1}^{T}\tilde{\mathbf{Y}}_{p}^{(n^*)^T}\tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)} - \boldsymbol{\Gamma}_p\boldsymbol{\mu}_{p-1} = 0. \tag{A.16}$$

Thus,

$$\boldsymbol{\mu}_{p-1} = 2\boldsymbol{\Gamma}_{p}^{-1}\cdot\mathbf{G}_{p-1}^{T}\tilde{\mathbf{Y}}_{p}^{(n^*)^T}\tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)}. \tag{A.17}$$

Since from (6.16) and (A.27),

$$\sum_{q=1}^{p-1}\mu_q\tilde{\mathbf{Y}}_{p}^{(n^*)}\mathbf{g}_q = \tilde{\mathbf{Y}}_{p}^{(n^*)}\mathbf{G}_{p-1}\boldsymbol{\mu}_{p-1}, \tag{A.18}$$

the equation (A.24) can be written as

$$\begin{aligned}
& 2\tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)} - 2\nu\mathbf{u}_{p}^{(n^*)} - \tilde{\mathbf{Y}}_{p}^{(n^*)}\mathbf{G}_{p-1}\boldsymbol{\mu}_{p-1} = 0 \\
\Rightarrow\quad & \nu\mathbf{u}_{p}^{(n^*)} = \tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)} - \tilde{\mathbf{Y}}_{p}^{(n^*)}\mathbf{G}_{p-1}\frac{\boldsymbol{\mu}_{p-1}}{2} \\
=\quad & \tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)} - \tilde{\mathbf{Y}}_{p}^{(n^*)}\mathbf{G}_{p-1}\boldsymbol{\Gamma}_{p}^{-1}\cdot\mathbf{G}_{p-1}^{T}\tilde{\mathbf{Y}}_{p}^{(n^*)^T}\tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)} \\
=\quad & \left[\mathbf{I}_{I_{n^*}} - \tilde{\mathbf{Y}}_{p}^{(n^*)}\mathbf{G}_{p-1}\boldsymbol{\Gamma}_{p}^{-1}\mathbf{G}_{p-1}^{T}\tilde{\mathbf{Y}}_{p}^{(n^*)^T}\right]\tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u}_{p}^{(n^*)}.
\end{aligned}$$

Using the definition in (6.15), an eigenvalue problem is obtained as $\boldsymbol{\Upsilon}_{p}^{(n^*)}\tilde{\mathbf{S}}_{T_p}^{(n^*)}\mathbf{u} = \nu\mathbf{u}$. Since ν is the criterion to be maximized, the maximization is achieved by setting $\mathbf{u}_{p}^{(n)^*}$ to be the (unit) eigenvector corresponding to the largest eigenvalue of (6.14). □

A.6 Proof of Corollary 5.1 in Chapter 5

Proof. To prove the corollary, it is only needed to show that for any mode n, the number of bases that can satisfy the zero-correlation constraint is upper-bounded by $\min\{I_n, M\}$.

Considering only one mode n, the zero-correlation constraint for mode $n^* = n$ in

(5.10) becomes,

$$\mathbf{u}_p^{(n)^T}\tilde{\mathbf{Y}}_p^{(n)}\mathbf{g}_q = 0, q = 1, ..., p-1. \tag{A.19}$$

First, let $\hat{\mathbf{g}}_p^{(n)^T} = \mathbf{u}_p^{(n)^T}\tilde{\mathbf{Y}}_p^{(n)} \in \mathbb{R}^{1\times M}$ and the constraint becomes

$$\hat{\mathbf{g}}_p^{(n)^T}\mathbf{g}_q = 0, q = 1, ..., p-1. \tag{A.20}$$

Since $\mathbf{g}_q \in \mathbb{R}^{M\times 1}$, when $p = M+1$, the set $\mathbf{g}_q, q = 1, ..., M$ forms a basis for the M-dimensional space and there is no solution for (A.20). Thus, $P \leq M$.

Second, let $\hat{\mathbf{u}}_q^{(n)} = \tilde{\mathbf{Y}}_p^{(n)}\mathbf{g}_q \in \mathbb{R}^{I_n\times 1}$ and the constraint becomes

$$\mathbf{u}_p^{(n)^T}\hat{\mathbf{u}}_q^{(n)} = 0, q = 1, ..., p-1. \tag{A.21}$$

Since $\mathbf{g}_q, q = 1, ..., p-1$ are orthogonal, $\hat{\mathbf{u}}_q^{(n)}, q = 1, ..., p-1$ are linearly independent if the elements of $\tilde{\mathbf{Y}}_p^{(n)}$ are not all zero. Since $\hat{\mathbf{u}}_q^{(n)} \in \mathbb{R}^{I_n\times 1}$, when $p = I_n + 1$, the set $\hat{\mathbf{u}}_q^{(n)}, q = 1, ..., p-1$ forms a basis for the I_n-dimensional space and there is no solution for (A.21). Thus, $P \leq I_n$.

From the above, $P \leq \min\{\min_n I_n, M\}$ if the elements of $\tilde{\mathbf{Y}}_p^{(n)}$ are not all zero, which is often the case as long as the projection basis is not initialized to zero and the elements of the training tensors are not all zero. □

A.7 Proof of Theorem 6.1 in Chapter 6

Proof. For a nonsingular $\tilde{\mathbf{S}}_{W_p}^{(n^*)}$, any $\mathbf{u}_p^{(n^*)}$ can be normalized such that

$$\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)}\mathbf{u}_p^{(n^*)} = 1 \tag{A.22}$$

and the ratio $\frac{\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)}}{\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)}\mathbf{u}_p^{(n^*)}}$ keeps unchanged. Therefore, the maximization of this ratio is equivalent to the maximization of $\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)}$ with the constraint (A.22). Lagrange

multipliers can be used to transform the problem (6.13) to the following to include all the constraints:

$$F(\mathbf{u}_p^{(n^*)}) = \mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - \nu\left(\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - 1\right) - \sum_{q=1}^{p-1}\mu_q\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{Y}}_p^{(n^*)}\mathbf{g}_q, \quad \text{(A.23)}$$

where ν and $\{\mu_q, q = 1, ..., p-1\}$ are Lagrange multipliers.

The optimization is performed by setting the partial derivative of $F(\mathbf{u}_p^{(n^*)})$ with respect to $\mathbf{u}_p^{(n^*)}$ to zero:

$$\frac{\partial F(\mathbf{u}_p^{(n^*)})}{\partial \mathbf{u}_p^{(n^*)}} = 2\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - 2\nu\tilde{\mathbf{S}}_{W_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - \sum_{q=1}^{p-1}\mu_q\tilde{\mathbf{Y}}_p^{(n^*)}\mathbf{g}_q = 0. \quad \text{(A.24)}$$

Multiplying (A.24) by $\mathbf{u}_p^{(n^*)^T}$ results in

$$2\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - 2\nu\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)}\mathbf{u}_p^{(n^*)} = 0 \Rightarrow \nu = \frac{\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)}}{\mathbf{u}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)}\mathbf{u}_p^{(n^*)}}, \quad \text{(A.25)}$$

which indicates that ν is exactly the criterion to be maximized.

Next, a set of $(p-1)$ equations are obtained through multiplying the equation (A.24) by $\mathbf{g}_q^T\tilde{\mathbf{Y}}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}$, $q = 1, ..., p-1$, respectively:

$$2\mathbf{g}_q^T\tilde{\mathbf{Y}}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - \sum_{q=1}^{p-1}\mu_q\mathbf{g}_q^T\tilde{\mathbf{Y}}_p^{(n^*)^T}\cdot\tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}\tilde{\mathbf{Y}}_p^{(n^*)}\mathbf{g}_q = 0. \quad \text{(A.26)}$$

Let

$$\boldsymbol{\mu}_{p-1} = [\mu_1\ \mu_2\ ...\ \mu_{p-1}]^T \quad \text{(A.27)}$$

and use (6.16), then the $(p-1)$ equations of (A.26) can be represented in a single matrix equation as following:

$$2\mathbf{G}_{p-1}^T\tilde{\mathbf{Y}}_p^{(n^*)^T}\tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}\tilde{\mathbf{S}}_{B_p}^{(n^*)}\mathbf{u}_p^{(n^*)} - \mathbf{G}_{p-1}^T\tilde{\mathbf{Y}}_p^{(n^*)^T}\cdot\tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}\tilde{\mathbf{Y}}_p^{(n^*)}\mathbf{G}_{p-1}\boldsymbol{\mu}_{p-1} = 0. \quad \text{(A.28)}$$

Thus,

$$\boldsymbol{\mu}_{p-1} = 2\left(\mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1}\right)^{-1} \cdot \mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)}. \quad \text{(A.29)}$$

Since from (6.16) and (A.27),

$$\sum_{q=1}^{p-1} \mu_q \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{g}_q = \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \boldsymbol{\mu}_{p-1}, \quad \text{(A.30)}$$

the equation (A.24) can be written as

$$\begin{aligned}
& 2\tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - 2\nu \tilde{\mathbf{S}}_{W_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \boldsymbol{\mu}_{p-1} = 0 \\
\Rightarrow \quad & \nu \tilde{\mathbf{S}}_{W_p}^{(n^*)} \mathbf{u}_p^{(n^*)} = \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \frac{\boldsymbol{\mu}_{p-1}}{2} \\
= \quad & \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)} - \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \left(\mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1}\right)^{-1} \mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)} \\
= \quad & \left[\mathbf{I}_{I_{n^*}} - \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1} \left(\mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}} \tilde{\mathbf{Y}}_p^{(n^*)} \mathbf{G}_{p-1}\right)^{-1} \mathbf{G}_{p-1}^T \tilde{\mathbf{Y}}_p^{(n^*)^T} \tilde{\mathbf{S}}_{W_p}^{(n^*)^{-1}}\right] \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u}_p^{(n^*)}
\end{aligned}$$

Using the definition in (6.15), a generalized eigenvalue problem is obtained as

$$\mathbf{R}_p^{(n^*)} \tilde{\mathbf{S}}_{B_p}^{(n^*)} \mathbf{u} = \nu \tilde{\mathbf{S}}_{W_p}^{(n^*)} \mathbf{u}. \quad \text{(A.31)}$$

Since ν is the criterion to be maximized, the maximization is achieved by setting $\mathbf{u}_p^{(n)^*}$ to be the (unit) generalized eigenvector corresponding to the largest generalized eigenvalue of (6.14). □

Appendix B

A Review on AdaBoost

Boosting is a general learning method that can be used in conjunction with many other learning algorithms to improve their performance. It is motivated by the question of whether a set of weak learners, which only performs slightly better than random guessing, can be boosted into an arbitrarily accurate strong learner [50,28]. Boosting produces a very accurate predication rule by combining rough and moderately accurate rules of thumb as finding many rough rules of thumb can be much easier than finding a single, highly accurate predication rule. The boosting algorithm starts with a weak learner that can find the rough rules of thumb. It then repeatedly calls this weak learner by feeding it a different subset of training samples [111]. Thus, each call generates a new weak predication rule. The boosting algorithm combines these weak rules into a single (hopefully) very accurate prediction rule [111]. It has been shown through both theoretical study and empirical testing that boosting is particularly robust in preventing overfitting and reducing the generalization error by increasing the so-called **margins** of the training examples [9, 111, 112]. The margin is defined as the minimal distance of an example to the decision surface of classification [129]. A larger expected margin of training data generally leads to a lower generalization error.

Among the many boosting algorithms, the AdaBoost formulated in [27] is a very

Input: A set of M training samples, $(\mathbf{y}_1, c_1), \cdots, (\mathbf{y}_M, c_M)$, $\mathbf{y}_m \in \mathbb{R}^I$, and $c_m \in \{-1, +1\}$.

Output: The final classifier $h(\mathbf{y})$.

Algorithm:

Initialize $D_1(m) = \frac{1}{M}$.

Do for $t = 1, ..., T$:

1. Train weak learner using the sample distribution D_t.
2. Get weak hypothesis $h_t : \mathbb{R}^I \to \{-1, +1\}$.
3. Calculate ϵ_t. If $\epsilon_t > 0.5$, stop (fail).
4. Choose $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$
5. Update:

$$\begin{aligned} D_{t+1}(m) &= \frac{D_t(m)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(\mathbf{y}_m) = c_m \\ e^{\alpha_t} & \text{if } h_t(\mathbf{y}_m) \neq c_m \end{cases} \\ &= \frac{D_t(m) \exp(-\alpha_t c_m h_t(\mathbf{y}_m))}{Z_t}, \end{aligned}$$

where Z_t is a normalization factor to ensure that D_{t+1} is a probability distribution.

Output: The final hypothesis:

$$h(\mathbf{y}) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{y}) \right)$$

Figure B.1: The AdaBoost algorithm.

popular one with great success [111, 120, 93]. The pseudo-code for AdaBoost is given in Fig. B.1 for vectorial input samples. The algorithm takes a training set of M samples $(\mathbf{y}_1, c_1)$,...,$((\mathbf{y}_M, c_M)$ as the input, where $\mathbf{y}_m \in \mathbb{R}^I$ and $c_m \in \{-1, +1\}$. It calls a weak learner repeatedly in a series of rounds $t = 1, ..., T$. In each call, a distribution (set of

weights) is maintained over the training set. The weight of this distribution on training sample $\mathbf{y}_m$ in round t is denoted by $D_t(m)$. All weights are initialized to be equal $D_1(m) = 1/M$ for $t = 1$.

In the boosting step t, the weak learner produces a weak hypothesis

$$h_t : \mathbb{R}^I \to \{-1, +1\} \tag{B.1}$$

for the distribution D_t. The goodness of h_t is then measured by the error ϵ_t defined as

$$\epsilon_t = \sum_{m:h_t(\mathbf{y}_m)\neq c_m} D_t(m). \tag{B.2}$$

Next, AdaBoost chooses a parameter α_t as in step 4 of Fig. B.1, which measures the importance of h_t. The distribution D_t is updated as in step 5 of Fig. B.1. The update effectively decreases the weights of those samples correctly classified by h_t and increases the weights of those classified incorrectly. In this way, the weak learner is forced to focus on the more difficult training samples in the next round [28]. Finally, the final strong hypothesis $h(\mathbf{y})$ is simply the weighted majority vote of all the weak hypothesis $h_t(\mathbf{y})$ as in Fig. B.1.

Bibliography

[1] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research*, vol. 1, pp. 113–141, Sep. 2000.

[2] 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.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[4] N. V. Boulgouris, D. Hatzinakos, and K. N. Plataniotis, "Gait recognition: a challenging signal processing technology for biometrics," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 78–90, Nov. 2005.

[5] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, "An angular transform of gait sequences for gait assisted recognition," in *Proc. IEEE International Conference on Image Processing*, vol. 2, Oct. 2004, pp. 857–860.

[6] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, "Gait recognition using linear time normalization," *Pattern Recognition*, vol. 39, no. 5, pp. 969–979, 2006.

[7] 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.

[8] J. E. Boyd, "Video phase-locked loops in gait recognition," in *Proc. IEEE Conference on Computer Vision*, vol. 1, Jul. 2001, pp. 696–703.

[9] L. Breiman, "Arcing classifiers," *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[10] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

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

[12] 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.

[13] R. Chellappa, A. Roy-Chowdhury, and S. Zhou, *Recognition of Humans and Their Activities Using Video*. Morgan & Claypool Publishers, San Rafael, California, 2005.

[14] A. Colombo, C. Cusano, and R. Schettini, "3D face detection using curvature analysis," *Pattern Recognition*, vol. 39, no. 3, pp. 444–455, Mar. 2006.

[15] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, pp. 287–314, 1994.

[16] P. Comon and B. Mourrain, "Decomposition of quantics in sums of powers of linear forms," *Signal Processing*, vol. 53, pp. 93–108, 1996.

[17] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[18] D. Cunado, M. S. Nixon, and J. N. Carter, "Automatic extraction and description of human gait models for recognition purposes," *Computer Vision and Image Understanding*, vol. 90, no. 1, pp. 1–41, Jan. 2003.

[19] C. M. Cyr and B. B. Kimia, "3D object recognition using shape similiarity-based aspect graph," in *Proc. IEEE Conference on Computer Vision*, vol. 1, Jul. 2001, pp. 254–261.

[20] G. Dai and D. Y. Yeung, "Tensor embedding methods," in *Proc. Twenty-First National Conference on Artificial Intelligence*, Jul. 2006, pp. 330–335.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley Interscience, 2001.

[22] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697–710, May 2002.

[23] N. M. Faber, R. Bro, and P. K. Hopke, "Recent developments in CANDECOMP/PARAFAC algorithms: a critical review," *Chemometrics and Intelligent Laboratory Systems*, vol. 65, no. 1, pp. 119–137, Jan. 2003.

[24] C. Faloutsos, T. G. Kolda, and J. Sun. (2007) Mining large time-evolving data using matrix and tensor tools. International Conference on Machine Learning

2007 Tutorial. [Online]. Available: http://www.cs.cmu.edu/~christos/TALKS/ICML-07-tutorial/ICMLtutorial.pdf

[25] J. P. Foster, M. S. Nixon, and A. Prügel-Bennett, "Automatic gait recognition using area-based metrics," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2489–2497, Oct. 2003.

[26] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. the Thirteenth International Conference on Machine Learning*, 1996, pp. 148–156.

[27] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[28] Y. Freund and R. E. Schapire, "A short introducion to boosting," *Journal of Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, 1999.

[29] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, Mar. 1989.

[30] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Boston, MA: Academic Press, 1990.

[31] M. G. Grant, J. D. Shutler, M. S. Nixon, and J. N. Carter, "Analysis of a human extraction system for deploying gait biometrics," in *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, Mar. 2004, pp. 46–50.

[32] R. D. Green and L. Guan, "Quantifying and recognizing human movement patterns from monocular video images-part II: applications to biometrics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 191–198, Feb. 2004.

[33] W. H. Greub, *Multilinear Algebra*. Berlin: Springer-Verlag, 1967.

[34] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 316–322, Feb. 2006.

[35] 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.

[36] X. He, "Incremental semi-supervised subspace learning for image retrieval," in *ACM conference on Multimedia 2004*, Oct. 2004, pp. 2–8.

[37] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems 18 (NIPS)*, 2005. [Online]. Available: http://books.nips.cc/papers/files/nips18/NIPS2005_0249.pdf

[38] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[39] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[40] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge Unverisity Press, 1985.

[41] G. Hua, P. A. Viola, and S. M. Drucker, "Face recognition using discriminatively trained orthogonal rank one tensor projections," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[42] A. K. Jain, R. Chellappa, S. C. Draper, N. Memon, P. J. Phillips, and A. Vetro, "Signal processing for biometric systems," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 146–152, Nov. 2007.

[43] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[44] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004.

[45] Z. Jin, J. Y. Yang, Z. S. Hu, and Z. Lou, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, vol. 34, pp. 1405–1416, 2001.

[46] Z. Jin, J. Y. Yang, Z. M. Tang, and Z. S. Hu, "A theorem on the uncorrelated optimal discriminant vectors," *Pattern Recognition*, vol. 34, no. 10, pp. 2041–2047, Oct. 2001.

[47] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer Serires in Statistics, 2002.

[48] A. Kale, "Algorithms for gait-based human identification from a monocular video sequences," Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Maryland College Park, 2003. [Online]. Available: http://www.cs.uky.edu/~amit/thesis.pdf

[49] A. Kale, A. N. Rajagopalan, A. Sunderesan, N. Cuntoor, A. Roy-Chowdhury, V. Krueger, and R. Chellappa, "Identification of humans using gait," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1163–1173, Sep. 2004.

[50] M. Kearns and L. G. Valiant, "Crytographic limitations on learning Boolean formulae and finite automata," *Journal of the Association for Computing Machinery*, vol. 41, no. 1, pp. 67–95, Jan. 1994.

[51] Y.-D. Kim and S. Choi, "Color face tensor factorization and slicing for illumination-robust recognition," in *Proc. International Conference on Biometrics*, August 2007, pp. 19–28.

[52] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[53] T. G. Kolda, "Orthogonal tensor decompositions," *SIAM Journal of Matrix Analysis and Applications*, vol. 23, no. 1, pp. 243–255, 2001.

[54] Y. Koren and L. Carmel, "Robust linear dimensionality reduction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 4, pp. 459–470, July-Aug. 2004.

[55] P. Kroonenberg, *Three-mode principal component analysis: theory and applications*. Leiden: DSWO Press, 1983.

[56] P. Kroonenberg and J. Leeuw, "Principal component analysis of three-mode data by means of alternating least squares algorithms," *Psychometrika*, vol. 45, no. 1, pp. 69–97, 1980.

[57] S. Lang, *Algebra*. Reading: Addison Wesley, 1984.

[58] L. D. Lathauwer, "Signal processing based on multilinear algebra," Ph.D. dissertation, Katholieke Universiteit Leuven, 1997. [Online]. Available: ftp://ftp.esat.kuleuven.ac.be/sista/delathauwer/reports/PHD.pdf

[59] 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.

[60] 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.

[61] 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.

[62] M. H. C. Law and A. K. Jain, "Incremental nonlinear dimensionality reduction by manifold learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 377–391, Mar. 2006.

[63] L. P. Lebedev and M. J. Cloud, *Tensor Analysis*. World Scientific, 2003.

[64] C. S. Lee and A. Elgammal, "Towards scalable view-invariant gait recognition: Multilinear analysis for gait," in *Proc. International Conference on Audio and Video-Based Biometric Person Authentication*, Jul. 2005, pp. 395–405.

[65] L. Lee, G. Dalley, and K. Tieu, "Learning pedestrian models for silhouette refinement," in *Proc. IEEE Conference on Computer Vision*, Oct. 2003, pp. 663–670.

[66] S. Z. Li, C. Zhao, M. Ao, and Z. Lei, "Learning to fuse 3D+2D based face recognition at both feature and decision levels," in *Proc. IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures*, Oct. 2005, pp. 43–53.

[67] S. Z. Li and A. K. Jain, "Introduction," in *Handbook of Face Recognition*, S. Z. Li and A. K. Jain, Eds. Springer-Verlag, 2004, pp. 1–11.

[68] J. J. Little and J. E.Boyd, "Recognizing people by their gait: the shape of motion," *Videre*, vol. 1, no. 2, pp. 1–32, 1998.

[69] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 725–737, May 2006.

[70] Q. Liu, X. Tang, H. Lu, and S. Ma, "Face recognition using kernel scatter-difference-based discriminant analysis," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 1081–1085, Jul. 2006.

[71] Y. Liu, R. T. Collins, and Y. Tsin, "A computational model for periodic pattern perception based on frieze and wallpaper groups," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 354–371, Mar. 2004.

[72] Z. Liu and S. Sarkar, "Simplest representation yet for gait recognition: averaged silhouette," in *Proc. International Conference on Pattern Recognition*, vol. 4, Aug. 2004, pp. 211–214.

[73] Z. Liu and S. Sarkar, "Effect of silhouette quality on hard problems in gait recognition," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 35, no. 2, pp. 170–178, 2005.

[74] M. Loog, R. P. W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 762–766, Jul. 2001.

[75] H. Lu, A. C. Kot, and Y. Q. Shi, "Distance-reciprocal distortion measure for binary document images," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 228–231, Feb. 2004.

[76] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Coarse-to-fine pedestrian localization and silhouette extraction for the gait challenge data sets," in *Proc. IEEE Conference on Multimedia and Expo*, Jul. 2006, pp. 1009–1012.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

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

[84] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition," *IEEE Transactions on Neural Networks*, 2008, accepted pending minor revision.

[85] 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*, Jul. 2008, pp. 616–623.

[86] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A taxonomy of emerging multilinear discriminant analysis solutions for biometric signal recognition," in *Biometrics: Theory, Methods, and Applications*, N. V. Boulgouris, K. Plataniotis, and E. Micheli-Tzanakou, Eds. Wiley/IEEE, 2009, to appear.

[87] 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.

[88] J. Lu, "Discriminant learning for face recognition," Ph.D. dissertation, University of Toronto, 2004. [Online]. Available: http://www.dsp.utoronto.ca/juwei/Publication/JuweiThesisUT04.pdf

[89] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 117–126, Jan. 2003.

[90] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using LDA based algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 195–200, Jan. 2003.

[91] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel discriminant learning with application to face recognition," in *Support Vector Machines: Theory and Applications*, L. WANG, Ed. Springer-Verlag, 2005, pp. 275–296.

[92] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition," *Pattern Recognition Letter*, vol. 26, no. 2, pp. 181–191, 2005.

[93] J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, and S. Z. Li, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 166–178, Jan. 2006.

[94] X. Lu, "Image analysis for face recognition," May 2003, 36 pages. [Online]. Available: http://www.face-rec.org/interesting-papers/General/ImAna4FacRcg_lu.pdf

[95] H. Moon and P. J. Phillips, "Computational and performance aspects of PCA-based face recognition algorithms," *Perception*, vol. 30, pp. 303–321, 2001.

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

[97] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[98] M. S. Nixon and J. N. Carter, "Advances in automatic gait recognition," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, May 2004, pp. 139–144.

[99] M. S. Nixon and J. N. Carter, "Automatic recognition by gait," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 2013–2024, Nov. 2006.

[100] C. Nolker and H. Ritter, "Visual recognition of continuous hand postures," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 983–994, Jul. 2002.

[101] P. J. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, , and W. Worek, "Overview of the face recognition grand challenge," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, Jun. 2005, pp. 947–954.

[102] P. J. Phillips, H. Moon, S. A. Rizvi, and P. Rauss, "The FERET evaluation method for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, Oct. 2000.

[103] J. L. Rodgers, W. A. Nicewander, and L. Toothaker, "Linearly independent, orthogonal and uncorrelated variables," *The American Statistician*, vol. 38, no. 2, pp. 133–134, May 1984.

[104] A. Ross and R. Govindarajan, "Feature level fusion of hand and face biometrics," in *Proc. SPIE Conference on Biometric Technology for Human Identification II*, Mar. 2005, pp. 196–204.

[105] A. Ross, A. K. Jain, and J. Z. Qian, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, pp. 2115–2125, 2003.

[106] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 22, pp. 2323–2326, Dec. 2000.

[107] H. S. Sahambi and K. Khorasani, "A neural-network appearance-based 3-D object recognition using independent component analysis," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 138–149, Jan. 2003.

[108] F. Samaria and S. Young, "HMM based architecture for face identification," *Image and Vision Computing*, vol. 12, pp. 537–583, 1994.

[109] S. Sarkar, P. J. Phillips, Z. Liu, I. Robledo, P. Grother, and K. W. Bowyer, "The human ID gait challenge problem: Data sets, performance, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 162–177, Feb. 2005.

[110] R. E. Schapire, "Using output codes to boost multiclass learning problems," in *Proc. the Fourteenth International Conference on Machine Learning*, 1997, pp. 313–321.

[111] R. E. Schapire, "The boosting approach to machine learning: An overview," in *MSRI Workshop on Nonlinear Estimation and Classification*, D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, and B. Yu, Eds. Springer, 2003. [Online]. Available: http://stat.haifa.ac.il/~goldensh/DM/msri.pdf

[112] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, "Boosting the margin: A new explanation for the effectiveness of voting methods," in *Proc. the Fourteenth International Conference on Machine Learning*, 1997, pp. 322–330.

[113] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, Dec. 1999.

[114] B. Schölkopf, A. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[115] 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.

[116] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, 2001, pp. 42–49.

[117] E. Shechtman and Y. C. ad M. Irani, "Space-time super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 531–545, Apr. 2005.

[118] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.

[119] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: 19 results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1948–1962, Nov. 2006.

[120] M. Skurichina and R. P. W. Duin, "Bagging, boosting and the random subspace method for linear classifiers," *Pattern Analysis & Applications*, vol. 5, no. 2, pp. 121–135, 2002.

[121] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: dynamic tensor analysis," in *Proc. the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, Aug. 2006, pp. 374–383.

[122] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos, "Less is more: Sparse graph mining with compact matrix decomposition," *Statistical Analysis and Data Mining*, vol. 1, no. 1, pp. 6–22, Feb. 2008.

[123] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Elapsed time in human gait recognition: A new approach," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, Apr. 2006, pp. 177 – 180.

[124] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[125] J. B. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 22, pp. 2319–2323, Dec. 2000.

[126] D. Tolliver and R. T. Collins, "Gait shape estimation for identification," in *Proc. International Conference on Audio and Video-Based Biometric Person Authentication*, Jun. 2003, pp. 734–742.

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

[128] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neurosicence*, vol. 3, no. 1, pp. 71–86, 1991.

[129] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.

[130] M. A. O. Vasilescu, "Human motion signatures: analysis, synthesis, recognition," in *Proc. International Conference on Pattern Recognition*, vol. 3, August 2002, pp. 456–460.

[131] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. seventh European Conference on Computer Vision*, May 2002, pp. 447–460.

[132] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *Proc. International Conference on Pattern Recognition*, vol. 2, August 2002, pp. 511–514.

[133] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear subspace analysis of image ensembles," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, June 2003, pp. 93–99.

[134] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear independent components analysis," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, June 2005, pp. 547–553.

[135] I. R. Vega and S. Sarkar, "Statistical motion model based on the change of feature relationships: human gait-based recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1323–1328, Oct. 2003.

[136] D. K. Wagg and M. S. Nixon, "On automated model-based extraction and analysis of gait," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, May 2004, pp. 11–16.

[137] J. Wang, H. Lu, K. N. Plataniotis, and J. Lu, "Gaussian kernel optimization for pattern classification," *Pattern Recognition*, submitted in 2008.

[138] J. Wang, K. N. Plataniotis, J. Lu, and A. N. Venetsanopoulos, "On solving the face recognition problem with one training sample per subject," *Pattern Recognition*, vol. 39, no. 9, pp. 1746–1762, 2006.

[139] J. Wang, K. N. Plataniotis, and A. N. Venetsanopoulos, "Selecting discriminant eigenfaces for face recognition," *Pattern Recognition Letters*, vol. 26, no. 10, pp. 1470–1482, 2005.

[140] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 149–158, Feb. 2004.

[141] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1505–1518, Dec. 2003.

[142] Y. Wang and S. Gong, "Tensor discriminant analysis for view-based object recognition," in *Proc. International Conference on Pattern Recognition*, vol. 3, August 2006, pp. 33 – 36.

[143] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, Jul. 1997.

[144] D. Xu, S. Lin, S. Yan, and X. Tang, "Rank-one projections with adaptive margins for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, vol. 37, no. 5, pp. 1226–1236, Oct. 2007.

[145] D. Xu, S. Yan, D. Tao, L. Zhang, X. Li, and H.-J. Zhang, "Human gait recognition with matrix representation," vol. 16, no. 7, pp. 896–903, Jul. 2006.

[146] D. Xu, S. Yan, L. Zhang, S. Lin, H.-J. Zhang, and T. S. Huang, "Reconstruction and recognition of tensor-based objects with concurrent subspaces analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 1, pp. 36–47, Jan. 2008.

[147] D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, and H.-Y. Shum, "Concurrent subspaces analysis," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. II, June 2005, pp. 203–208.

[148] R. Xu and D. W. II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.

[149] C. Y. Yam, M. S. Nixon, and J. N. Carter, "Automated person recognition by walking and running via model-based approaches," *Pattern Recognition*, vol. 37, no. 5, pp. 1057–1072, May 2004.

[150] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 212–220, Jan. 2007.

[151] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[152] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Discriminant analysis with tensor representation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, June 2005, pp. 526–532.

[153] J. Yang, D. Zhang, A. F. Frangi, and J. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[154] 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.

[155] J. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, no. 1-3, pp. 167–191, 2005.

[156] J. Ye, R. Janardan, and Q. Li, "GPCA: An efficient dimension reduction scheme for image compression and retrieval," in *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 354–363.

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

[158] J. Ye, R. Janardan, Q. Li, and H. Park, "Feature reduction via generalized uncorrelated linear discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1312–1322, Oct. 2006.

[159] 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.

[160] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, 2003.