Multilinear Algebra in High-Order Data Analysis: Retrieval, Classification and Representation

BY

QUN LI
B.S., Nanjing University of Science and Technology, Nanjing, China, 2009
M.S., University of Illinois at Chicago, Chicago, IL, 2012

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2013

Chicago, Illinois

Defense Committee:

Dan Schonfeld, Chair and Advisor
Rashid Ansari
Natasha Devroye
Daniela Tuninetti
Shmuel Friedland, Math, Stats, & Computer Sci

To my family.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Dan Schonfeld for his vision, wisdom, guidance and encouragement during my PhD research that will benefit me more in my future life.

I would also like to thank Prof. Rashid Ansari for his concern and efforts that support me throughout my PhD study. It is also my honor to work with Prof. Shmuel Friedland and learn from his serious attitude in scientific research. I would also like to extend my gratitude to my dissertation committee members Prof. Daniela Tuninetti and Prof. Natasha Devroye for valuable discussions and suggestions on shaping my thesis.

Finally, I would like to thank my family, for their support at all times.

QL

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HOSVD	Higher-Order Singular Value Decomposition
SVD	Singular Value Decomposition
CAVIAR	Context Aware Vision using Image-based Active Recognition
LDA	Linear Discriminant Analysis
MDA	Multilinear Discriminant Analysis
TTP	Tensor-to-Tensor Projection
TVP	Tensor-to-Vector Projection
GTDA	General Tensor Discriminant Analysis
DATER	Discriminant Analysis with Tensor Representation
DGTDA	Direct General Tensor Discriminant Analysis
CMDA	Constrained Multilinear Discriminant Analysis
FDA	Fisher Discriminant Analysis
2DFDA	Two-Dimensional Fisher Discriminant Analysis
2DLDA	Two-Dimensional Linear Discriminant Analysis
MDA	Multilinear Discriminant Analysis

LIST OF ABBREVIATIONS (Continued)

TensorLDA	Tensor Linear Discriminant Analysis
TR1DA	Tensor Rank-One Discriminant Analysis
UMLDA	Uncorrelated Multilinear Discriminant Analysis
CS	Compressive Sensing
GTCS	Generalized Tensor Compressive Sensing
GTCS-S	Generalized Tensor Compressive Sensing–Serial
GTCS-P	Generalized Tensor Compressive Sensing–Parallelizable
NSP	Null Space Property
RIP	Restricted Isometry Property
BCS	Block-Based Compressive Sensing
KCS	Kronecker Compressive Sensing
PSNR	Peak Signal to Noise Ratio.

SUMMARY

One of the fundamental problems in data analysis is how to represent the data. Real-world signals of practical interest such as color imaging, video sequences and multi-sensor networks, are usually generated by the interaction of multiple factors and thus can be intrinsically represented by higher-order tensors. Application of conventional linear analysis methods to higher-order data tensor representation is typically performed by conversion of the data to very long vectors, thus inevitably losing spatial locality as well as imposing a huge computational and memory burden. As a result, great efforts have been made to extend conventional linear analysis methods that rely on data representation in the form of vectors, for higher-order data analysis. This thesis is dedicated to the study of higher-order data analysis including retrieval, classification and representation, within the mathematical framework provided by multilinear algebra.

We first present a higher-order singular value decomposition (HOSVD)-based method for robust indexing and retrieval of higher-order data in responding to various query structures. We prove theoretically that, for real tensors, the set of HOSVD orthogonal matrices of a subtensor is equivalent to the corresponding subset of HOSVD orthogonal matrices of the original tensor. Therefore, if we first arrange all tensors in the database compactly as a higher-order tensor, then we only need to conduct HOSVD once on the total tensor.

We then extend linear discriminant analysis (LDA) for higher-order data classification. We propose two multilinear discriminant analysis methods, Direct General Tensor Discrimi-

nant Analysis (DGTDA) and Constrained Multilinear Discriminant Analysis (CMDA). Both DGTDA and CMDA seek a tensor-to-tensor projection onto a lower-dimensional tensor subspace, which is most efficient for discrimination.

Finally, we propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order tensors. GTCS offers an efficient means for representation of multidimensional data by providing simultaneous acquisition and compression from all tensor modes.

CHAPTER 1

INTRODUCTION

One of the fundamental problems in data analysis is how to represent the data. While conventional data analysis methods such as linear discriminant analysis (LDA) and compressive sensing (CS) theory rely on data representation in the form of vectors, many data types in various applications such as color imaging, video sequences, and multi-sensor networks, are intrinsically represented by higher-order tensors. Application of conventional analysis methods to higher-order data representation is typically performed by conversion of the data to very long vector, thus inevitably losing spatial locality and imposing a huge computational and memory burden. Recently, multilinear algebra, the algebra of higher-order tensors, was applied to the analysis of the multi-factor structure of multidimensional signals. Tensor defines multilinear operators over a set of vector spaces and is a natural generalization of vector and matrix. Consequently, multilinear analysis subsumes linear analysis as a special case and offers a unifying mathematical framework to address problems involving multi-factor data.

This thesis is dedicated to the study of high-order data analysis including retrieval, classification and representation, within the mathematical framework provided by multilinear algebra.

In Chapter 2, we first prove that for real tensors, the HOSVD orthogonal matrices of a subtensor can be well approximated by those corresponding mode matrices obtained from applying HOSVD to the original tensor. We then propose a robust HOSVD-based multilinear approach for efficiently indexing and retrieving multifactor data according to the format of the query,

either complete or partial. Simulation in the context of multi-object multi-camera motion trajectory indexing and retrieval demonstrated the efficiency and robustness of the proposed approach.

In Chapter 3, we first show that a closed-form solution to the optimal projection sought by GTDA exists. We subsequently propose Direct GTDA (DGTDA) which not only gets rid of parameter tuning but also achieves the optimal projection directly. We demonstrate that DGTDA outperforms GTDA in terms of both training efficiency and classification accuracy. In addition, we propose Constrained Multilinear Discriminant Analysis (CMDA) that looks for a set of projection matrices with orthonormal columns by iteratively maximizing the scatter ratio criterion. We prove theoretically that in the limit, the value of the scatter ratio criterion in CMDA approaches its extreme value, if it exists, with bounded error. In fact, experimental results show that in most cases, the optimization procedure of CMDA converges, thus leading to superior and stabler classification performance in comparison to DATER. To our best knowledge, CMDA is the first scatter ratio maximization-based MDA method that exhibits convergency.

Finally in Chapter 4, we propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order tensors. GTCS offers an efficient means for representation of multidimensional data by providing simultaneous acquisition and compression from all tensor modes. In addition, we compare the performance of the proposed method with Kronecker compressive sensing (KCS). We demonstrate experimentally that GTCS outperforms KCS in terms of both accuracy and speed.

CHAPTER 2

HIGHER-ORDER SINGULAR VALUE DECOMPOSITION (HOSVD)-BASED HIGH-ORDER DATA INDEXING AND RETRIEVAL

Higher-order singular value decomposition (HOSVD), a natural multilinear extension of the matrix SVD, computes the orthonormal spaces associated with different modes of the tensor. It is widely employed for feature extraction, dimensionality reduction etc. However, due to the vast quantities of tensor entries involved in calculation, it inevitably suffers from high computational cost, especially when recalculation of HOSVD is frequently required. To address the problem, we prove theoretically that for real tensors, the set of HOSVD orthogonal matrices of a sub-tensor is equivalent to the corresponding subset of HOSVD orthogonal matrices of the original tensor. Therefore, if we first arrange all tensors in the database compactly as a higher-order tensor, then we only need to conduct HOSVD once on the total tensor. We subsequently propose a robust HOSVD-based multilinear approach for efficiently indexing and retrieving multifactor data, in responding to various query structures. We also apply the proposed method for indexing and retrieval of multi-camera multi-object motion trajectory. Simulation results demonstrate the superior performance of the proposed approach in terms of both robustness and efficiency.

2.1 Introduction

Singular value decomposition (SVD) has served as a powerful tool in linear analysis. To perform SVD, samples should be represented in vector form where only one factor is allowed to

vary. However, in most real applications, the data are the composite consequence of multiple factors and naturally require higher-order tensor representation. For instance, Shashua and Levin (1) first employed 3^{rd}-order tensor instead of matrix of vectorized images to represent an image ensemble. Correspondingly, matrix SVD needs to be generalized in order to deal with multifactor data. The higher-order singular value decomposition (HOSVD) is such a natural extension of matrix SVD within the mathematical framework of multilinear algebra. Although it has been shown that some nice properties of SVD such as uniqueness and existence cannot be guaranteed in its higher-order counterpart, HOSVD still provides satisfactory performance in multilinear analysis. For example, Alex et al. (2) demonstrated the power of HOSVD in the context of facial image ensemble classification.

Specifically in the field of motion analysis, such as motion trajectory indexing and retrieval, large volume motion data are usually represented compactly in higher-order tensor form. One common problem is that these tensors are usually of very high dimensionality and therefore, to accelerate processing, dimensionality reduction is always necessary. In analogy to applying SVD for feature extraction as well as dimensionality reduction of linear samples, HOSVD offers a multilinear tool to seek the optimal lower-dimensional tensor subspace that preserves the data class structure. Due to the vast quantities of tensor entries involved in calculation, HOSVD inevitably suffers from high computational cost and therefore is less employed in applications where recalculation of HOSVD is frequently needed. For instance, referring to the motion trajectory indexing and retrieval problem mentioned above, if the structure of the query is known and fixed, all the samples in the database can be stored in the same structure as the

query and HOSVD needs only to be conducted once. However, in practice, the query may contain either complete information as the database do or only partial information from certain modes. In the sense of tensor, a partial query may be (1) either a tensor of the same order yet of smaller size in certain modes or (2) a lower-order sub-tensor. Apparently, obtaining the HOSVD unitary matrices of the new tensors efficiently, preferably without recalculating HOSVD will greatly improve the efficiency of the processing algorithms. In (3), Xiang et al. presented dynamic tensor HOSVD downdating algorithm to address the first problem and this paper will focus on the solution to the latter.

The rest of the paper is organized as follows. We first prove theoretically in Section 2.2 that the HOSVD unitary matrices of a sub-tensor can be well approximated by those corresponding mode matrices obtained from applying HOSVD to the original tensor. We then propose a robust HOSVD-based multilinear approach for efficiently indexing and retrieving multifactor data according to the format of the query in Section 2.3. Section 2.4 applies the proposed method to multifactor motion trajectory analysis which can dynamically adjust the database according to the query structure. At last, Section 3.7 concludes the paper and discusses briefly on future work.

2.2 Theoretical Foundation

We propose Theorem 2.2.3 which later serves as the theoretical foundation of the higher-order tensor data indexing and retrieval algorithm introduced in Section 2.3. In order to prove Theorem 2.2.3, we state two lemmas first.

Lemma 2.2.1 *(4, Thm. 3.3) Let* $\mathcal{X} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$. *Let* $U_k \in \mathbb{R}^{L_k \times L_k^{'}}$ *where* $U_k^T U_k = I$ *and* $L_k^{'} \leq L_k$ *for* $k = 1, \ldots, M$. *Then the function* $f(\widehat{\mathcal{X}}) = \|\mathcal{X} - \widehat{\mathcal{X}}\|_F^2$, *where* $rank_k(\widehat{\mathcal{X}}) = L_k^{'}$, *is minimized, when* $\widehat{\mathcal{X}} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$ *is given by* $\widehat{\mathcal{X}} = (\mathcal{X} \prod_{k=1}^{M} \times_k U_k^T) \prod_{k=1}^{M} \times_k U_k = \mathcal{X} \prod_{k=1}^{M} \times_k (U_k U_k^T)$.

Proof It is sufficient to prove $\mathcal{Y}$ minimizes $g(\mathcal{Y}) = \|\mathcal{X} - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k\|_F^2$, where $\mathcal{Y} = \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T \in \mathbb{R}^{L_1' \times \ldots \times L_M'}$.

Let $\mathcal{E} = \mathcal{X} - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k$. Then we have

$$\begin{aligned}
\mathcal{E} \prod_{k=1}^{M} \times_k U_k^T &= (\mathcal{X} - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k) \prod_{k=1}^{M} \times_k U_k^T \\
&= \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k \prod_{k=1}^{M} \times_k U_k^T \\
&= \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T - \mathcal{Y} \prod_{k=1}^{M} \times_k (U_k^T U_k) \\
&= \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T - \mathcal{Y}.
\end{aligned}$$

Therefore, $\mathcal{Y} \prod_{k=1}^{M} \times_k U_k$ is the least square estimation of $\mathcal{X}$, i.e. $g(\mathcal{Y})$ is minimized when $\mathcal{Y} = \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T$. This completes the proof.

Lemma 2.2.2 *(4, Thm. 3.4) Let* $\mathcal{X} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$. *Let* $U_k \in \mathbb{R}^{L_k \times L_k^{'}}$ *where* $U_k^T U_k = I$ *and* $L_k^{'} \leq L_k$ *for* $k = 1, \ldots, M$. *Then maximizing* $f(U_k|_{k=1}^{M}) = \|\mathcal{X} - \mathcal{X} \prod_{k=1}^{M} \times_k (U_k U_k^T)\|_F^2$ *is equivalent to minimizing* $g(U_k|_{k=1}^{M}) = \|\mathcal{X} \prod_{k=1}^{M} \times_k U_k^T\|_F^2$.

Proof Let $\mathcal{Y} = \mathcal{X}\prod_{k=1}^{M}\times_k U_k^T \in \mathbb{R}^{L'_1\times\ldots\times L'_M}$.

$$\begin{aligned} f(U_k|_{k=1}^M) &= \|\mathcal{X} - (\mathcal{X}\prod_{k=1}^{M}\times_k U_k^T)\prod_{k=1}^{M}\times_k U_k\|_F^2 \\ &= \|\mathcal{X} - \mathcal{Y}\prod_{k=1}^{M}\times_k U_k\|_F^2 \\ &= \|\mathcal{X}\|_F^2 - 2\langle\mathcal{X}, (\mathcal{X}\prod_{k=1}^{M}\times_k U_k^T)\prod_{k=1}^{M}\times_k U_k\rangle + \|\mathcal{Y}\|_F^2 \\ &= \|\mathcal{X}\|_F^2 - 2\langle\mathcal{X}\prod_{k=1}^{M}\times_k U_k^T, \mathcal{X}\prod_{k=1}^{M}\times_k U_k^T\rangle + \|\mathcal{Y}\|_F^2 \\ &= \|\mathcal{X}\|_F^2 - \|\mathcal{Y}\|_F^2. \end{aligned}$$

Thus maximizing $f(U_k|_{k=1}^M)$ is equivalent to minimizing $g(U_k|_{k=1}^M) = \|\mathcal{Y}\|_F^2$. This completes the proof.

Theorem 2.2.3 *Let $\mathcal{X} \in \mathbb{R}^{L_1\times\ldots\times L_M}$. For an arbitrary set $\{i_1,\ldots,i_n\} \subset \{1,\ldots,M\}$, let $(U_{i_1}^*,\ldots,U_{i_n}^*)$ be a solution of*

$$(U_{i_1}^*,\ldots,U_{i_n}^*) = \arg\min_{(U_{i_1},\ldots,U_{i_n})}\sum_{j=1}^{J}\|\mathcal{X}_j - \mathcal{Y}_j\prod_{k=1}^{n}\times_{i_k}U_{i_k}\|_F^2,$$

$$J = L_{i_{n+1}}\cdot\ldots\cdot L_{i_M}, \{i_{n+1},\ldots,i_M\} = \{1,\ldots,M\}\backslash\{i_1,\ldots,i_n\}, \tag{2.1}$$

where $U_{i_k}^* \in \mathbb{R}^{L_{i_k}\times L'_{i_k}}$, $U_{i_k}^{*T}U_{i_k}^* = I$, $L'_{i_k} \le L_{i_k}$ for $k = 1,\ldots,n$. $\mathcal{Y}_j \in \mathbb{R}^{L'_{i_1}\times\ldots\times L'_{i_n}}$, $\mathcal{X}_j \in \mathbb{R}^{L_{i_1}\times\ldots\times L_{i_n}}$ is the j^{th} sub-tensor of $\mathcal{X}$ obtained by varying indices $i_1,\ldots,i_n$ with fixed indices $i_{n+1},\ldots,i_M$ and $rank_k(\mathcal{Y}_j\prod_{k=1}^{n}\times_{i_k}U_{i_k}) = L'_k$.

Let $(U_1^*, \ldots, U_M^*)$ *be a solution of*

$$(U_1^*, \ldots, U_M^*) = \arg \min_{(U_1,\ldots,U_M)} \|\mathcal{X} - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k\|_F^2, \tag{2.2}$$

where $U_{i_k}^* \in \mathbb{R}^{L_{i_k} \times L'_{i_k}}$, $U_{i_k}^{*T} U_{i_k}^* = I$ *for* $k = 1, \ldots, M$. $L'_{i_k} \leq L_{i_k}$ *for* $k = 1, \ldots, n$ *and* $L'_{i_k} = L_{i_k}$ *for* $k = n+1, \ldots, M$. $\mathcal{Y} \in \mathbb{R}^{L'_1 \times \ldots \times L'_M}$ *and* $rank_k(\mathcal{Y} \prod_{k=1}^{M} \times_k U_k) = L'_k$.

Then $(U_{i_1}^*, \ldots, U_{i_n}^*)$ *in* (Equation 2.1) *is equivalent to the* $(U_{i_1}^*, \ldots, U_{i_n}^*)$ *tuple of* $(U_1^*, \ldots, U_M^*)$ *in* (Equation 2.2).

Proof According to Lemma 2.2.1, $\|\mathcal{X} - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k\|_F^2$ is minimized when $\mathcal{Y} = \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T \in \mathbb{R}^{L'_1 \times \ldots \times L'_M}$. Since for $k = n+1, \ldots, M$, $L'_{i_k} = L_{i_k}$, we also have $U_{i_k}^* U_{i_k}^{*T} = I$. Therefore, according to Lemma 2.2.2

$$\begin{aligned}
&\min_{(U_1,\ldots,U_M)} \|\mathcal{X} - \mathcal{Y} \prod_{k=1}^{M} \times_k U_k\|_F^2 \\
&= \|\mathcal{X} - \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T \prod_{k=1}^{M} \times_k U_k\|_F^2 \\
&= \|\mathcal{X} - \mathcal{X} \prod_{k=1}^{M} \times_k (U_k U_k^T)\|_F^2 \\
&= \|\mathcal{X} - \mathcal{X} \prod_{k=1}^{n} \times_{i_k} (U_{i_k} U_{i_k}^T)\|_F^2 \\
&= \min_{(U_{i_1},\ldots,U_{i_n})} \|\mathcal{X} - \mathcal{X} \prod_{k=1}^{n} \times_{i_k} U_{i_k}^T \prod_{k=1}^{n} \times_{i_k} U_{i_k}\|_F^2 \\
&= \max_{(U_{i_1},\ldots,U_{i_n})} \|\mathcal{X} \prod_{k=1}^{n} \times_{i_k} U_{i_k}^T\|_F^2.
\end{aligned}$$

Hence $\|\mathcal{X} \prod_{k=1}^{n} \times_{i_k} U_{i_k}^T\|_F^2$ is maximized by the $(U_{i_1}^*, \ldots, U_{i_n}^*)$ tuple of $(U_1^*, \ldots, U_M^*)$.

Next, let $\mathcal{B} = \mathcal{X}\prod_{k=2}^{n}\times_{i_k} U_{i_k}^T$. Thus

$$\begin{aligned}
\mathcal{B}_{(i_1)} &= \mathcal{X}_{(i_1)}(U_{i_n} \otimes \ldots \otimes U_{i_2}) \\
&= [\mathcal{X}_{1(i_1)} \cdots \mathcal{X}_{J(i_1)}](U_{i_n} \otimes \ldots \otimes U_{i_2}) \\
&= [\mathcal{X}_{1(i_1)}(U_{i_n} \otimes \ldots \otimes U_{i_2}) \cdots \mathcal{X}_{J(i_1)}(U_{i_n} \otimes \ldots \otimes U_{i_2})],
\end{aligned}$$

where $J = L_{i_{n+1}} \cdot \ldots \cdot L_{i_M}$ and $\mathcal{X}_{j(i_1)} \in \mathbb{R}^{L_{i_1} \times (L_{i_2} \cdot \ldots \cdot L_{i_n})}$ for $j = 1, \ldots, J$. Then we have

$$\begin{aligned}
(U_{i_1}^*, \ldots, U_{i_n}^*) &= \arg \max_{(U_{i_1}, \ldots, U_{i_n})} \|\mathcal{X}\prod_{k=1}^{n}\times_{i_k} U_{i_k}^T\|_F^2 \\
&= \arg \max_{(U_{i_1}, \ldots, U_{i_n})} \|\mathcal{B}\times_{i_1} U_{i_1}^T\|_F^2 \\
&= \arg \max_{(U_{i_1}, \ldots, U_{i_n})} tr\{U_{i_1}^T \mathcal{B}_{(i_1)} \mathcal{B}_{(i_1)}^T U_{i_1}\} \\
&= \arg \max_{(U_{i_1}, \ldots, U_{i_n})} tr\{U_{i_1}^T [\sum_{j=1}^{J} \mathcal{X}_{j(i_1)}(U_{i_n} \otimes \ldots \otimes U_{i_2}) \\
&\quad (U_{i_n} \otimes \ldots \otimes U_{i_2})^T \mathcal{X}_{j(i_1)}^T] U_{i_1}\} \\
&= \arg \max_{(U_{i_1}, \ldots, U_{i_n})} \sum_{j=1}^{J} \|\mathcal{X}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}\|_F^2 \\
&= \arg \min_{(U_{i_1}, \ldots, U_{i_n})} \sum_{j=1}^{J} \|\mathcal{X}_j - \mathcal{Y}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}\|_F^2.
\end{aligned}$$

This completes the proof.

Suppose the HOSVD of a real tensor $\mathcal{X} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$ is

$$\mathcal{X} = \mathcal{S} \prod_{k=1}^{M} \times_k U_k,$$

where $U_k \in \mathbb{R}^{L_k \times L_k}$'s are orthogonal and $\mathcal{S} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$. In other words, $\mathcal{S} = \mathcal{X} \prod_{k=1}^{M} \times_k U_k^T$ minimizes $\|\mathcal{X} - \mathcal{S} \prod_{k=1}^{M} \times_k U_k\|_F^2 = 0$. Based on Theorem 2.2.3, the $(U_{i_1}, \ldots, U_{i_n})$ tuple of $(U_1, \ldots, U_M)$ also minimizes $\sum_{j=1}^{J} \|\mathcal{X}_j - \mathcal{Y}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}\|_F^2 = 0$, which means each $\|\mathcal{X}_j - \mathcal{Y}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}\|_F^2 = 0$. Therefore,

$$\mathcal{X}_j = \mathcal{Y}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}, \quad j = 1, \ldots, (L_{i_{n+1}} \cdot \ldots \cdot L_{i_M}).$$

That is to say, $U_{i_1}, \ldots, U_{i_n}$ also serve as the HOSVD orthogonal matrices of the sub-tensor $\mathcal{X}_j$. Also, the corresponding core tensor $\mathcal{Y}_j$ can be obtained by $\mathcal{X}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}^T$. Once the HOSVD of the total tensor is known, there is no need to calculate the HOSVD of any sub-tensors, which in fact can be obtained directly.

2.3 Higher-Order Data Indexing and Retrieval

Assume that the whole database is represented compactly as an M^{th}-order tensor $\mathcal{X} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$, where different tensor modes correspond to different factors of the data. Given an arbitrary query tensor $\mathcal{T} \in \mathbb{R}^{L_{i_1} \times \ldots \times L_{i_n}}$ consisting of a subset $\{i_1, \ldots, i_n\}$ of the M tensor modes. The higher-order data indexing procedure and retrieval procedure are summarized in Table Table I and Table II respectively. Although $\mathcal{S} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$, it usually contains vast zero entries, thus can serve as indexing tensors efficiently.

TABLE I

THE HIGHER-ORDER DATA INDEXING PROCEDURE

Input	The total database tensor $\mathcal{X} \in \mathbb{R}^{L_1 \times \ldots \times L_M}$, the tensor modes $\{i_1, \ldots, i_n\}$ contained in the query tensor.
1.	Conduct HOSVD on the database tensor $\mathcal{X}$ and obtain unitary projection matrices $U_1, \ldots, U_M$ and the core tensor $\mathcal{S}$, i.e. $\mathcal{X} = \mathcal{S}\prod_{k=1}^{M} \times_k U_k$.
2.	Rearrange $\mathcal{X}$ into sub-tensors $\mathcal{X}_j \in \mathbb{R}^{L_{i_1} \times \ldots \times L_{i_n}}$ for $j = 1, \ldots, (L_{i_{n+1}} \cdot \ldots \cdot L_{i_M})$.
3.	Obtain indexing tensor $\mathcal{X}_j^{ind}$ by $\mathcal{X}_j^{ind} = \mathcal{X}_j \prod_{k=1}^{n} \times_{i_k} U_{i_k}^T$, for $j = 1, \ldots, (L_{i_{n+1}} \cdot \ldots \cdot L_{i_M})$.
Output	Indexing tensors $\mathcal{X}_j^{ind}$ for $j = 1, \ldots, (L_{i_{n+1}} \cdot \ldots \cdot L_{i_M})$.

TABLE II

THE HIGHER-ORDER DATA RETRIEVAL PROCEDURE

Input	The query tensor $\mathcal{T} \in \mathbb{R}^{L_{i_1} \times \ldots \times L_{i_n}}$ and indexing tensors $\mathcal{X}_j^{ind}$ for $j = 1, \ldots, (L_{i_{n+1}} \cdot \ldots \cdot L_{i_M})$, the similarity threshold σ.
1.	Obtain the indexing query tensor,i.e. $\mathcal{T}^{ind} = \mathcal{T} \prod_{k=1}^{n} \times_{i_k} U_{i_k}^T$.
2.	Compare the Frobenius distance D_j between indexing tensor $\mathcal{X}_j^{ind}$ and $\mathcal{T}^{ind}$, i.e. $D_j = \|\mathcal{X}_j^{ind} - \mathcal{T}^{ind}\|_F$, retrieve those whose D_j is less than σ.
Output	Retrieved sub-tensors $\mathcal{X}_j$'s.

2.4 Simulation Results

We test the performance of the proposed approach on the 2^{nd} CAVIAR dataset (5) for the indexing and retrieval of multi-camera multi-object motion trajectories.

2.4.1 Tensor representation of the 2^{nd} CAVIAR data set

The 2^{nd} CAVIAR dataset contains a set of surveillance video clips of the same scene obtained by cameras from two different viewpoints, one corridor view and one frontal view. We first concatenate x- and y-location information of each object trajectory into one column vector, called single trajectory vector $\mathbf{v}$. We then align $\mathbf{v}$'s as columns of multiple trajectory matrix M whose column number is equal to the number of objects in the particular video sequence. Here, each multiple trajectory matrix contains the motion information of a group of objects within one video clip. Multiple trajectory matrices with the same number of columns are then aligned to form a 3^{rd}-order tensor, referred to as multiple trajectory tensor. Finally, multiple trajectory tensors from different cameras construct a 4^{th}-order tensor $\mathcal{X}$ along the dimension of cameras. Figure Figure 1 gives an example of the process to obtain a 4^{th}-order tensor from 2 cameras each containing 3 video sequences of time duration L with 2 moving objects. For video sequences of different length, every single trajectory is sampled to the same length $2L$.

Similarly, we form the CAVIAR database into a $200 \times 2 \times 47 \times 2$ tensor with modes corresponding to motion trajectory, object, video clip and camera respectively. To be more specific, this tensor consists of 47 video clips obtained by 2 cameras with 2 motion trajectories of length 200 in each clip.

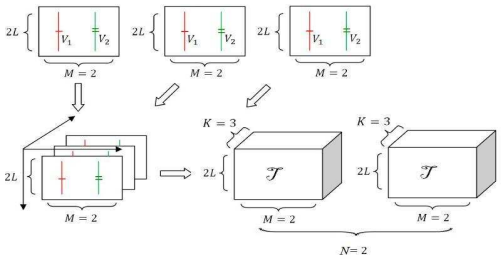


Figure 1. Tensor representation of multi-camera multi-object motion trajectories.

2.4.2 Complete query—two trajectories from two cameras

We first input a complete query of size $200 \times 2 \times 2$ consisting of motion trajectory matrices from two cameras. The precision and recall curve in this case is depicted in Figure Figure 2(a). Retrieval results are shown in Figure Figure 3.

(a) (b)

Figure 2. Precision and recall curve: (a) complete query; (b) partial query.

Figure 3. Retrieval results for complete query: (a)-(b) query; (c)-(d) most-similar results; (e)-(f) second most-similar results.

2.4.3 Partial query 1–single trajectory from two cameras

In this case, we select one single motion trajectory from each camera and form a partial query of size 200×2. The retrieval results and precision and recall curve are shown in Figure Figure 4, Figure Figure 2(b) respectively.

2.4.4 Partial query 2–double trajectories from one camera

In the last case, we select one motion trajectory matrix of size 200×2 from one camera as a different kind of partial query. The retrieval results are shown in Figure Figure 5.

To sum up, the experimental results demonstrate the efficiency and robustness of the proposed approach in responding to various query structures.

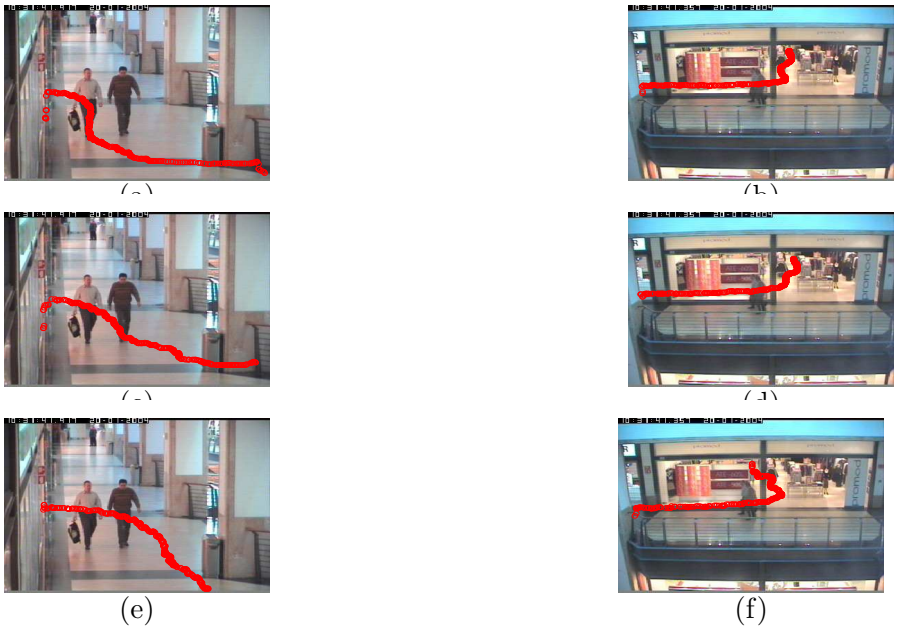


Figure 4. Retrieval results for partial query 1: (a)-(b) query; (c)-(d) most-similar results; (e)-(d) second most-similar results.

2.5 Conclusion

In this paper, we first proved theoretically that the set of HOSVD unitary matrices of a sub-tensor is equivalent to the corresponding subset of HOSVD unitary matrices of the original tensor. We then proposed a robust HOSVD-based multilinear approach for efficiently indexing and retrieving multifactor data, in responding to various query structures. Simulation results demonstrated the robustness and the superior performance of the proposed approach. Although we only applied our approach to motion trajectory analysis, it can serve as a unifying framework for a variety of computer vision problems involving multifactor data. As can be seen in the simulation part, we assumed that the certain modes contained in the query were known and

(a)

(b)

(c)

Figure 5. Retrieval results for partial query case 2: (a) query; (b) most-similar results; (c) second most-similar results.

adjusted the database accordingly. However, in most real applications, the correspondence between the tensor structure in the query and database is unknown. Future work will focus on the solution to the unknown case.

CHAPTER 3

MULTILINEAR DISCRIMINANT ANALYSIS FOR HIGHER-ORDER TENSOR DATA CLASSIFICATION

Linear discriminant analysis (LDA) has played a crucial role as a subspace learning method in computer vision and pattern recognition applications. LDA relies on data representation in the form of vectors. However, many data types do not lend themselves to vector representation. Instead, real-world data, usually generated from the interaction of multiple factors, can be naturally represented by higher-order tensors. Recent efforts have been made to extend LDA for tensor data classification, which is generally referred to as the multilinear discriminant analysis (MDA) problem. MDA seeks a tensor-to-tensor projection (TTP) to a lower-dimensional tensor subspace that is most efficient for discrimination (some literature aim at a tensor-to-vector projection (TVP), which is beyond the scope of this paper). Existing examples include General Tensor Discriminant Analysis (GTDA) and Discriminant Analysis with Tensor Representation (DATER). To measure the separation of samples in the new tensor subspace, MDA methods mainly employ one of the two criteria: scatter ratio criterion (e.g. DATER) and scatter difference criterion (e.g. GTDA). The optimal TTP should be the one that maximizes such criterion. Due to the dependency among tensor modes, it seems that no closed-form solution to this optimization problem exists, hence both the two methods attempt to resolve such dependency through iterative approximation. GTDA is known to be the first MDA method that converges over iterations. However, its performance relies highly on tuning of the parameter in the scatter

difference criterion. On the other hand, although DATER usually results in better classification performance, it does not converge, yet the number of iterations executed upon termination has a direct impact on DATER's performance. We first show that a closed-form solution to the optimal projection sought by GTDA exists. We subsequently propose Direct GTDA (DGTDA) which not only gets rid of parameter tuning but also achieves the optimal projection directly. We demonstrate that DGTDA outperforms GTDA in terms of both training efficiency and classification accuracy. In addition, we propose Constrained Multilinear Discriminant Analysis (CMDA) that looks for a set of projection matrices with orthonormal columns by iteratively maximizing the scatter ratio criterion. We prove theoretically that in the limit, the value of the scatter ratio criterion in CMDA approaches its extreme value, if it exists, with bounded error. In fact, experimental results show that in most cases, the optimization procedure of CMDA converges, thus leading to superior and stabler classification performance in comparison to DATER. To our best knowledge, CMDA is the first scatter ratio maximization-based MDA method that exhibits convergency.

3.1 Introduction

Linear discriminant analysis (LDA)(6) has been widely employed for subspace learning (e.g. dimensionality reduction, feature extraction etc.) in computer vision and pattern recognition applications. Although real data of natural and social sciences are usually of very high dimension, the underlying structure can in many cases be characterized by a small number of parameters. For instance, in many statistical pattern recognition problems, such as face

recognition (7)(8) and image retrieval (9), in order to visualize the intrinsic structure of the high-dimensional data, LDA often serves as a preprocessing step to reduce the dimensionality.

One of the fundamental problems in data analysis is how to represent the data. Image is intrinsically a matrix. However, since LDA takes vectors as input, image typically has to be vectorized first, during which spatial locality is inevitably lost. Some recent works have started to consider an image object as a matrix for unsupervised learning problem(10; 11). Many efforts have been devoted to the extension of LDA which takes matrices as input. Liu et al. (12) proposed a special LDA that projects matrix data to some vector space for discrimination. Later, Kong et al. (13) extended traditional Fisher Discriminant Analysis (FDA) to 2DFDA where data matrix is projected onto a two-dimensional tensor subspace and showed its advantages in solving small sample size problem. Its multi-class counterpart 2DLDA was then proposed by Ye et al. (14). Due to the dependency of the projection matrices on each other, no direct solutions exist. Therefore, they derived an iterative algorithm that fixes one of the projection matrices at a time. However, 2DLDA does not converge over iterations. Similar method was employed by (15) for tensor subspace learning. Whereas in (16), TensorLDA overcomes such dependency by imposing an orthonormality constraint on the two matrices and arrived at closed-form solutions. However, such solutions are not optimal.

In fact, natural images are generated by the interaction of multiple factors related to scene structure, illumination and imaging. Recently, multilinear algebra, the algebra of higher-order tensors, was applied to the analysis of the multi-factor structure of image ensembles (17; 18; 19; 20). Tensor defines multilinear operators over a set of vector spaces and is a nat-

ural generalization of matrix. Consequently, multilinear analysis subsumes linear analysis as a special case and offers a unifying mathematical framework to address problems involving multi-factor data. Vasilescu and Terzopoulos presented Tensorface (17) which represents a set of face images as a higher-order tensor and applies higher-order singular value decomposition (HOSVD) to disentangle the constituent factors. However, since Tensorface still considers each image as a vector, it is computationally expensive and not optimal for recognition. Our previous work (20) also employed HOSVD for dimensionality reduction, yet we represented each image as a matrix and achieved lower retrieval error rate than Tensorface.

Within multilinear algebra framework, the extension of LDA for tensor data classification has gained growing interest over the past few years, which is usually referred to as multilinear discriminant analysis (MDA) problem. Generally speaking, MDA methods can be categorized into two directions based on dimensionality of the learned subspace (21): MDA that seeks a tensor-to-vector projection (TVP) for discrimination in a lower-dimensional vector space and MDA that looks for a tensor-to-tensor projection (TTP) for discrimination in a tensor subspace. To measure the separation of samples in the new tensor subspace, two criteria are usually employed: the scatter ratio criterion and the scatter difference criterion. The optimal projection should be the one that maximizes such criterion. The first TVP-based MDA was known as Tensor Rank-One Discriminant Analysis (TR1DA) (22; 23), derived from tensor rank-one decomposition (24). TR1DA aims at a TVP that maximizes the scalar scatter difference criterion. However, due to this criterion, TR1DA relies on coordinates. Later, another TVP-based MDA method, Uncorrelated Multilinear Discriminant Analysis (UMLDA) was introduced

in (25). UMLDA extracts uncorrelated discriminative features through a TVP that optimizes a scalar scatter ratio criterion. In terms of TTP-based MDA methods, the discriminant analysis with tensor representation (DATER)(18), later also known as Multilinear Discriminant Analysis (MDA)(26) was proposed for tensor data classification (to avoid confusion, we refer to this method as DATER in following discussions). However, like its 2D counterpart 2DLDA(14), DATER does not converge over iterations either. Thus it is hard to determine the number of iterations it should run before termination, which has a direct impact on DATER's performance. On the other hand, (19) proposed General Tensor Discriminant Analysis (GTDA) which learns a tensor subspace by scatter difference maximization. GTDA is known to be the first convergent MDA method. However, its performance relies highly on tuning of the parameter in the scatter difference criterion. In this paper, we first show that a closed-form solution to the optimal projection sought by GTDA exists. We subsequently propose Direct GTDA (DGTDA) which not only gets rid of parameter tuning but also achieves the optimal projection directly. We demonstrate that DGTDA outperforms GTDA in terms of both training time efficiency and classification accuracy. In addition, we propose Constrained Multilinear Discriminant Analysis (CMDA) that looks for a set of projection matrices with orthonormal columns by iteratively maximizing the scatter ratio criterion. We prove theoretically that in the limit, the value of the scatter ratio criterion in CMDA approaches its extreme value, if it exists, with bounded error.

Our main contributions are summarized as follows.

1. We prove mathematically the existence of a global maximum of the scatter difference criterion that GTDA attempts to optimize over iterations and as a matter of fact, such

global maximum can be obtained directly. We then propose a closed-form solution to GTDA, namely, DGTDA.

2. The proposed DGTDA also gets rid of parameter tuning which has a major impact on the performance of GTDA. We show that DGTDA outperforms GTDA in terms of both training efficiency and classification accuracy.

3. We propose CMDA which learns a set of projection matrices with orthonormal columns by iteratively maximizing the scatter ratio criterion. We prove theoretically that in the limit, the value of the scatter ratio criterion in CMDA approaches its extreme value, if it exists, with bounded error. In fact, experimental results show that in most cases, the optimization procedure of CMDA converges, thus leading to superior and stabler classification performance in comparison to DATER. To our best knowledge, CMDA is the first scatter ratio maximization-based MDA method that exhibits convergency.

4. We also show that unlike DGTDA to GTDA, no closed-form solution exists to avoid the iterative procedure in CMDA.

The rest of the chapter is organized as follows. Section 3.2 first reviews basic concepts in multilinear algebra and LDA. Section 3.3 generalizes LDA concepts to their multilinear counterparts. Section 3.4 then proposes DGTDA and CMDA. Section 3.5 introduces a simple nearest neighbor classifier employed in the simulations. Section 3.6 experimentally compares DGTDA and CMDA to GTDA and DATER. Finally, Section 3.7 concludes the paper.

3.2 Review of Multilinear Algebra and Linear Discriminant Analysis

3.2.1 Multilinear Algebra Background

In this section, we introduce the following definitions frequently used in multilinear algebra (27) as well as notations used in this paper. Throughout the discussion, lower-case characters represent scalar values $(a, b, \ldots)$, bold-face characters represent vectors $(\mathbf{a}, \mathbf{b}, \ldots)$, capitals represent matrices $(A, B, \ldots)$ and calligraphic capitals represent tensors $(\mathcal{A}, \mathcal{B}, \ldots)$.

A tensor is a multidimensional array. The order of a tensor is the number of tensor modes. For instance, tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ has order N and the dimension of its n^{th} mode (also called mode n directly) is I_n.

Kronecker Product The Kronecker product of matrices $A \in \mathbb{R}^{I \times J}$ and $B \in \mathbb{R}^{K \times L}$ is denoted by $A \otimes B$. The result is a matrix of size$(IK) \times (JL)$ and defined by

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1J}B \\ a_{21}B & a_{22}B & \cdots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}B & a_{I2}B & \cdots & a_{IJ}B \end{pmatrix}.$$

Two useful properties of Kronecker product are: $(A \otimes B)(C \otimes D) = AC \otimes BD$ and $(A \otimes B)^T = A^T \otimes B^T$.

Mode-n Product The mode-n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ and a matrix $U \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n U$ and is of size $I_1 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$. By element, we have $(\mathcal{X} \times_n U)_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}$.

Mode-n Fiber and Mode-n Unfolding The mode-n fiber of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is obtained by fixing every index but i_n. The mode-n unfolding of $\mathcal{X}$, which is also called mode-n matricization, is denoted by $X_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting $I_n \times (I_1 \cdot I_2 \cdot \ldots \cdot I_{n-1} \cdot I_{n+1} \cdot \ldots \cdot I_N)$ matrix.

We have, $\mathcal{Y} = \mathcal{X} \times_1 U_1 \times_2 U_2 \times \ldots \times_N U_N \Leftrightarrow Y_{(n)} = U_n X_{(n)} (U_N \otimes \ldots \otimes U_{n+1} \otimes U_{n-1} \otimes \ldots \otimes U_1)^T$.

To simplify the notation, we denote $\mathcal{X} \times_1 U_1 \times_2 U_2 \times \ldots \times_N U_N$ by $\mathcal{X} \prod_{k=1}^{N} \times_k U_k$ and denote $U_N \otimes \ldots \otimes U_{n+1} \otimes U_{n-1} \otimes \ldots \otimes U_1$ by $\otimes_{k=N, k \neq n}^{1} U_k$.

3.2.2 Linear Discriminant Analysis Background

Linear Discriminant Analysis (LDA) seeks the direction of projection that is most efficient for discrimination in a lower-dimensional subspace. Suppose that we have a set of p d-dimensional vector samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ belonging to c classes and n_i is the number of samples in class i such that $p = \sum_{i=1}^{c} n_i$. Let $\mathbf{x}_{i,j}$ denote the j^{th} sample in class i, then the mean vector of class i is given by $\mathbf{m}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{i,j}$ and the total mean vector of all samples is $\mathbf{m} = \frac{1}{p} \sum_{i=1}^{c} \sum_{j=1}^{n_i} \mathbf{x}_{i,j} = \frac{1}{p} \sum_{i=1}^{c} n_i \mathbf{m}_i$. In Fisher Discriminant Analysis (FDA)(6) where $c = 2$, only one discriminant function is needed. Hence a natural generalization for $c-$class problem involves $c-1$ discriminant functions. Therefore, our goal is to find a projection from d-dimensional space to a $(c-1)-$dimensional subspace. Expressed in matrix form, $\mathbf{y} = U^T \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{y} \in \mathbb{R}^{c-1}$ and $U = [\mathbf{u}_1 \ \mathbf{u}_2 \cdots \mathbf{u}_{c-1}] \in \mathbb{R}^{d \times (c-1)}$.

A measure of separation between two projected classes is the distance between the projected sample means. For instance, square of the distance between projected sample means of class a and b is, $\|U^T \mathbf{m}_a - U^T \mathbf{m}_b\|_F^2 = \|U^T (\mathbf{m}_a - \mathbf{m}_b)\|_F^2 = tr\{U^T (\mathbf{m}_a - \mathbf{m}_b)(\mathbf{m}_a - \mathbf{m}_b)^T U\}$, where the

subscript F stands for Frobenius norm. The larger the distance is, the better the separation between class a and b is.

To avoid the trivial case where we enlarge this distance by merely scaling U, we can maximize it relative to some measure of the standard deviation within classes. If we define the scatter matrix S_i for class i and the within-class scatter matrix S_W by $S_i = \sum_{j=1}^{n_i}(\mathbf{x}_{i,j}-\mathbf{m}_i)(\mathbf{x}_{i,j}-\mathbf{m}_i)^T$ and $S_W = \sum_{i=1}^{c} S_i$ respectively, then an estimate of the standard deviation of the projected samples in class i, denoted by $\widetilde{S}_i$ can be expressed as

$$\begin{aligned}\widetilde{S}_i &= \sum_{j=1}^{n_i}(U^T\mathbf{x}_{i,j} - U^T\mathbf{m}_i)(U^T\mathbf{x}_{i,j} - U^T\mathbf{m}_i)^T \\ &= U^T[\sum_{j=1}^{n_i}(\mathbf{x}_{i,j} - \mathbf{m}_i)(\mathbf{x}_{i,j} - \mathbf{m}_i)^T]U \\ &= U^T S_i U, \end{aligned} \tag{3.1}$$

and the within-class scatter matrix in the projected subspace $\widetilde{S_W}$ is

$$\begin{aligned}\widetilde{S_W} &= \sum_{i=1}^{c}\sum_{j=1}^{n_i}(U^T\mathbf{x}_{i,j} - U^T\mathbf{m}_i)(U^T\mathbf{x}_{i,j} - U^T\mathbf{m}_i)^T \\ &= \sum_{i=1}^{c} U^T S_i U = U^T(\sum_{i=1}^{c} S_i)U, \\ &= U^T S_W U.\end{aligned}$$

Similarly, if we define the between-class scatter matrix S_B, an estimate of the standard deviation between classes, to be $S_B = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T$, then in the projected subspace,

$$\begin{aligned}\widetilde{S_B} &= \sum_{i=1}^{c} n_i(U^T\mathbf{m}_i - U^T\mathbf{m})(U^T\mathbf{m}_i - U^T\mathbf{m})^T \\ &= U^T[\sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T]U \\ &= U^T S_B U.\end{aligned}$$

Now our goal is to find the optimal projection to a lower-dimensional space that maximizes the between-class scatter $\widetilde{S_B}$ while minimizing the within-class scatter $\widetilde{S_W}$. Therefore, the objective function that we try to optimize is,

$$J(U) = \frac{\| \sum_{i=1}^{c} n_i U^T(\mathbf{m}_i - \mathbf{m})\|^2}{\| \sum_{i=1}^{c} \sum_{j=1}^{n_i} U^T(\mathbf{x}_{i,j} - \mathbf{m}_i)\|^2} = \frac{tr\{U^T S_B U\}}{tr\{U^T S_W U\}}. \tag{3.2}$$

(Equation 3.2) is usually referred to as the scatter ratio criterion. Another criterion that is also frequently used is the scatter difference criterion defined as,

$$J(U) = tr\{U^T S_B U\} - \zeta tr\{U^T S_W U\}. \tag{3.3}$$

The solution to (Equation 3.3) is equivalent to that to (Equation 3.2) when ζ in (Equation 3.3) is the Lagrange multiplier. It is known that (Equation 3.2) can be solved as the generalized eigenvalue problem, that is, the first n column vectors of an optimal U are the generalized

eigenvectors that correspond to the n largest eigenvalues in $S_B\mathbf{u} = \lambda S_W\mathbf{u}$ (28). Moreover, if S_W is nonsingular, this can be converted to a conventional eigenvalue problem.

3.3 Multilinear Discriminant Analysis: Objective

In this section, we extend LDA concepts to their counterparts in the framework of multilinear algebra. Suppose that we have a set of p tensor samples $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_p \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ belonging to c classes and again n_i is the number of samples in class i such that $p = \sum_{i=1}^{c} n_i$. Let $\mathcal{X}_{i,j}$ denote the j^{th} sample in class i, then the class mean tensor for class i is given by $\mathcal{M}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathcal{X}_{i,j}$ and the total mean tensor is $\mathcal{M} = \frac{1}{p}\sum_{i=1}^{c}\sum_{j=1}^{n_i} \mathcal{X}_{i,j} = \frac{1}{p}\sum_{i=1}^{c} n_i\mathcal{M}_i$.

The projection now is from a $I_1 \times I_2 \times \ldots \times I_N$-dimensional tensor space to a $I'_1 \times I'_2 \times \ldots \times I'_N$-dimensional tensor subspace. Our goal is to find the set of optimal projection matrices $U_1, U_2, \ldots, U_N$ ($U_n \in \mathbb{R}^{I_n \times I'_n}$ for $n = 1, \ldots, N$) for the most accurate classification in the projected subspace where

$$\mathcal{Y}_{i,j} = \mathcal{X}_{i,j} \prod_{k=1}^{N} \times_k U_k^T \in \mathbb{R}^{I'_1 \times I'_2 \times \ldots \times I'_N}. \tag{3.4}$$

Then the sample mean for projected class i is given by

$$\begin{aligned} \widetilde{\mathcal{M}}_i &= \frac{1}{n_i}\sum_{j=1}^{n_i} \mathcal{Y}_{i,j} = \frac{1}{n_i}\sum_{j=1}^{n_i} (\mathcal{X}_{i,j} \prod_{k=1}^{N} \times_k U_k^T) \\ &= \mathcal{M}_i \prod_{k=1}^{N} \times_k U_k^T \end{aligned} \tag{3.5}$$

and is simply the projection of $\mathcal{M}_i$. Similarly, the total mean tensor of the projected samples is $\mathcal{M}\prod_{k=1}^{N}\times_k U_k^T$.

As in LDA, distances between the c projected sample means serve as our measure of separation for the projected samples. We employ Frobenius norm of the difference between two tensors to measure the separation or distance between the two tensors. As a result, it does not matter if we calculate the distance through tensors or their matrix unfoldings. Therefore, we may convert tensors to the more familiar matrix form by matrix unfolding.

In stead of using tensor samples, if we first unfold the tensors to mode-n unfoldings and view the unfolded matrices as our training samples, then the mode-n between-class scatter matrix in the projected, by all tensor modes, tensor subspace, is defined as follows,

$$
\begin{aligned}
B_n =& \sum_{i=1}^{c} n_i[(\mathcal{M}_i\prod_{k=1}^{N}\times_k U_k^T)_{(n)} - (\mathcal{M}\prod_{k=1}^{N}\times_k U_k^T)_{(n)}][(\mathcal{M}_i\prod_{k=1}^{N}\times_k U_k^T)_{(n)} - (\mathcal{M}\prod_{k=1}^{N}\times_k U_k^T)_{(n)}]^T \\
=& \sum_{i=1}^{c} n_i[(\mathcal{M}_i - \mathcal{M})\prod_{k=1}^{N}\times_k U_k^T]_{(n)}[(\mathcal{M}_i - \mathcal{M})\prod_{k=1}^{N}\times_k U_k^T]_{(n)}^T \\
=& \sum_{i=1}^{c} n_i[U_n^T(\mathcal{M}_i - \mathcal{M})_{(n)}(\otimes_{k=N,k\neq n}^{1} U_k^T)^T][U_n^T(\mathcal{M}_i - \mathcal{M})_{(n)}(\otimes_{k=N,k\neq n}^{1} U_k^T)^T]^T \\
=& U_n^T[\sum_{i=1}^{c} n_i(\mathcal{M}_i - \mathcal{M})_{(n)}(\otimes_{k=N,k\neq n}^{1} U_k)(\otimes_{k=N,k\neq n}^{1} U_k^T)(\mathcal{M}_i - \mathcal{M})_{(n)}^T]U_n \\
=& U_n^T\{\sum_{i=1}^{c} n_i[(\mathcal{M}_i - \mathcal{M})\prod_{k=1,k\neq n}^{N}\times_k U_k^T]_{(n)}[(\mathcal{M}_i - \mathcal{M})\prod_{k=1,k\neq n}^{N}\times_k U_k^T]_{(n)}^T\}U_n \\
=& U_n^T B_n^{\bar{n}} U_n.
\end{aligned}
$$

Here, $B_n^{\bar{n}}$ denotes the mode-n between-class scatter matrix in the projected, by all tensor modes except for mode n, tensor subspace, where the subscript n specifies scatter in terms of mode-n

unfolded samples and the superscript $\bar{n}$ specifies the non-projection mode n, in other words, the tensor samples are projected by all tensor modes, except for mode n. The mode-n between class scatter matrix characterizes separation between c classes in terms of mode-n unfoldings of the tensor samples.

Similarly, the mode-n within-class scatter matrix is defined as,

$$\begin{aligned}
W_n =& \sum_{i=1}^{c}\sum_{j=1}^{n_i}[(\mathcal{X}_{i,j}\prod_{k=1}^{N}\times_k U_k^T)_{(n)} - (\mathcal{M}_i\prod_{k=1}^{N}\times_k U_k^T)_{(n)}][(\mathcal{X}_{i,j}\prod_{k=1}^{N}\times_k U_k^T)_{(n)} - (\mathcal{M}_i\prod_{k=1}^{N}\times_k U_k^T)_{(n)}]^T \\
=& U_n^T\{\sum_{i=1}^{c}\sum_{j=1}^{n_i}[(\mathcal{X}_{i,j}-\mathcal{M}_i)\prod_{k=1,k\neq n}^{N}\times_k U_k^T]_{(n)}[(\mathcal{X}_{i,j}-\mathcal{M}_i)\prod_{k=1,k\neq n}^{N}\times_k U_k^T]_{(n)}^T\}U_n \\
=& U_n^T W_n^{\bar{n}} U_n,
\end{aligned}$$

where $W_n^{\bar{n}}$ represents the mode-n within-class scatter matrix in the projected, except for mode n, tensor subspace.

Employing the above definition, maximizing $\sum_{i=1}^{c} n_i\|(\mathcal{M}_i-\mathcal{M})\prod_{k=1}^{N}\times_k U_k^T\|_F^2$, the Frobenius distance between the projected sample means, is equivalent to maximizing $tr\{U_n^T B_n^{\bar{n}} U_n\}$, simply because of the fact that $\|A\|_F^2 = tr(AA^T)$ for any matrix A and that matrix unfolding does not affect the Frobenius norm. Similarly, minimizing $\sum_{i=1}^{c}\sum_{j=1}^{n_i}\|(\mathcal{X}_{i,j}-\mathcal{M}_i)\prod_{k=1}^{N}\times_k U_k^T\|_F^2$

is equivalent to minimizing $tr\{U_n^T W_n^{\bar{n}} U_n\}$. As a result, for each mode n, we have an objective function

$$J(U_n) = \frac{\sum_{i=1}^{c} n_i \|(\mathcal{M}_i - \mathcal{M}) \prod_{k=1}^{N} \times_k U_k^T\|_F^2}{\sum_{i=1}^{c} \sum_{j=1}^{n_i} \|(\mathcal{X}_{i,j} - \mathcal{M}_i) \prod_{k=1}^{N} \times_k U_k^T\|_F^2} \tag{3.6}$$

$$= \frac{tr\{U_n^T B_n^{\bar{n}} U_n\}}{tr\{U_n^T W_n^{\bar{n}} U_n\}}, \tag{3.7}$$

and the set of optimal projection matrices should maximize $J(U_n)$ for $n = 1, \ldots, N$ simultaneously to best preserve the given class structure.

Although obtained through different derivations, it can be easily shown that the S_B, S_W in DATER(18) as well as B_n, W_n in GTDA (19) are in fact the same as $B_n^{\bar{n}}$, $W_n^{\bar{n}}$, thus we denote them by $B_n^{\bar{n}}$, $W_n^{\bar{n}}$ uniformly throughout following discussions. DATER has the same objective function for each mode n as in (Equation 3.7) while GTDA optimizes a generalized scatter difference criterion of (Equation 3.3), that is, $tr(U_n^T B_n^{\bar{n}} U_n) - \zeta tr(U_n^T W_n^{\bar{n}} U_n)$ under the constraint that $U_n^T U_n = I$, where ζ is a user controlled tuning parameter.

3.4 Multilinear Discriminant Analysis: Algorithm

Because the projection matrices for each mode depend on those of the other modes, they cannot be computed independently. One common approach is to employ iterative approximation. Existing examples include DATER and GTDA. We begin with a review of each algorithm and we then introduce the proposed algorithms.

3.4.1 Direct General Tensor Discriminant Analysis

Table Table III summarizes the training procedure of GTDA. At step 5, GTDA seeks to

TABLE III

TRAINING PROCEDURE OF GTDA

Input:	Training tensors $\mathcal{X}_{i,j}\,|_{1\leq i\leq c}^{1\leq j\leq n_i} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, their class labels $i\in\{1,2,\ldots,c\}$, dimensionality of the reduced tensor subspace $I'_1\times I'_2\times\ldots\times I'_N$, the tuning parameter ζ, the maximum number of iterations T_{max}.
Initialization:	Initialize $U_n^0 = 1_{I_n\times I'_n}\,|_{n=1}^{N}$ (All entries of U_n^0 are 1).
Step 1.	For $t=1,2,\ldots,T_{max}\{$
Step 2.	For $n=1,2,\ldots,N$ do$\{$
Step 3.	Calculate $B_n^{\bar{n}t}=\sum_{i=1}^{c} n_i[(\mathcal{M}_i-\mathcal{M})\times_1 U_1^{tT}\times\ldots\times_{n-1}U_{n-1}^{tT}$ $\times_{n+1}U_{n+1}^{t-1T}\times\ldots\times_N U_N^{t-1T}]_{(n)}[(\mathcal{M}_i-\mathcal{M})\times_1 U_1^{tT}\times\ldots$ $\times_{n-1}U_{n-1}^{tT}\times_{n+1}U_{n+1}^{t-1T}\times\ldots\times_N U_N^{t-1T}]_{(n)}^T$;
Step 4.	Calculate $W_n^{\bar{n}t}=\sum_{i=1}^{c}\sum_{j=1}^{n_i}[(\mathcal{X}_{i,j}-\mathcal{M}_i)\times_1 U_1^{tT}\times\ldots$ $\times_{n-1}U_{n-1}^{tT}\times_{n+1}U_{n+1}^{t-1T}\times\ldots\times_N U_N^{t-1T}]_{(n)}[(\mathcal{X}_{i,j}-\mathcal{M}_i)\times_1 U_1^{tT}$ $\times\ldots\times_{n-1}U_{n-1}^{tT}\times_{n+1}U_{n+1}^{t-1T}\times\ldots\times_N U_N^{t-1T}]_{(n)}^T$;
Step 5.	Optimize $U_n^{\star t}=\arg\max_{U^TU=I} tr[U^T(B_n^{\bar{n}t}-\zeta W_n^{\bar{n}t})U]$ by SVD on $B_n^{\bar{n}t}-\zeta W_n^{\bar{n}t}$. $\}$End For loop in Step 2.
Step 6.	Check convergence: $Err(t)=\sum_{n=1}^{N}\|U_n^tU_n^{t-1T}-I\|_F\leq\varepsilon$. $\}$End For loop in Step 1.
Step 7.	$\mathcal{Y}_{i,j}=\mathcal{X}_{i,j}\prod_{k=1}^{N}\times_k U_k^T$.
Output:	The projection matrices $U_n\in\mathbb{R}^{I_n\times I'_n}\,|_{n=1}^{N}$ constrained by $U_n^TU_n=I$ and the projected tensors $\mathcal{Y}_{i,j}\,|_{1\leq i\leq c}^{1\leq j\leq n_i}\in\mathbb{R}^{I'_1\times I'_2\times\ldots\times I'_N}$.

optimize $tr[U_n^T(B_n^{\bar{n}} - \zeta W_n^{\bar{n}})U_n]$ with the constraint $U_n^T U_n = I$ by singular value decomposition (SVD) on $B_n^{\bar{n}} - \zeta W_n^{\bar{n}}$. We show next that a global optimum of this objective can be obtained directly without iterations and subsequently propose an efficient algorithm, Direct General Tensor Discriminant Analysis (DGTDA).

Theorem 3.4.1 *The global optimum of GTDA objective function can be obtained directly.*

Proof For simplicity, we prove the 2^{nd} order case, where samples are 2^{nd} order tensors, namely matrices$\in \mathbb{R}^{I_1 \times I_2}$. Note that the proof can be easily extended to higher-order case. Now we aim to find 2 projection matrices $U_1 \in \mathbb{R}^{I_1 \times I_1'}$ and $U_2 \in \mathbb{R}^{I_2 \times I_2'}$. To avoid confusion, we first introduce the notations used in the proof:

$$
\begin{aligned}
B_1^{\bar{1},\bar{2}} &= \sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(1)} (\mathcal{M}_i - \mathcal{M})_{(1)}^T, \\
W_1^{\bar{1},\bar{2}} &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(1)} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(1)}^T, \\
B_2^{\bar{1},\bar{2}} &= \sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (\mathcal{M}_i - \mathcal{M})_{(2)}^T, \\
W_2^{\bar{1},\bar{2}} &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T, \\
B_2^{\bar{2}} &= \sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} U_1 U_1^T (\mathcal{M}_i - \mathcal{M})_{(2)}^T, \\
W_2^{\bar{2}} &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} U_1 U_1^T (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T,
\end{aligned}
$$

where the subscript specifies the unfolded mode and the superscripts specify the modes without projection.

For the 2^{nd} order case, apparently we have the following relationship:

$$\begin{aligned}
B_1^{\bar{1},\bar{2}} &= \sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(1)} (\mathcal{M}_i - \mathcal{M})_{(1)}^T \\
&= \sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)}^T (\mathcal{M}_i - \mathcal{M})_{(2)} \\
W_1^{\bar{1},\bar{2}} &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(1)} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(1)}^T \\
&= \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}.
\end{aligned}$$

Suppose at iteration t, we obtain U_1^t by SVD on some matrix. Let us denote by $\mathbf{u}_{1,k}^t$ the k^{th} column of this U_1^t and keep only the first I_1' columns so that $U_1^t \in \mathbb{R}^{I_1 \times I_1'}$ now. Therefore, $U_1^t U_1^{tT} = \sum_{k=1}^{I_1'} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT} = I - \sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}$. Here, we use $\mathbf{u}_{1,k}^{tT}$ to denote the transpose of $\mathbf{u}_{1,k}^t$, that is, $(\mathbf{u}_{1,k}^t)^T$ to save space later. Consequently, at iteration t, the optimal U_2^t can then be obtained by

$$\begin{aligned}
U_2^{\star t} &= \arg \max_{U_2^T U_2 = I} tr[U_2^T (B_2^{\bar{2}t} - \zeta W_2^{\bar{2}t}) U_2] \\
&= \arg \max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} U_1^t U_1^{tT} (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2\} \\ -\zeta tr\{U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} U_1^t U_1^{tT} \\ (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2\} \end{array} \right\} \\
&= \arg \max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (I - \sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) \\ (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2\} - \zeta tr\{U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ (I - \sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2\} \end{array} \right\}
\end{aligned}$$

$$
\begin{aligned}
&= \arg\max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2\} - \\ tr\{U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (\sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) \\ (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2\} - \zeta tr\{U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2\} + \zeta tr\{U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ (\sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2\} \end{array} \right\} \\
&= \arg\max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T (B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}) U_2\} - tr\{U_2^T \\ \left\{ \begin{array}{l} [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (\sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) \\ (\mathcal{M}_i - \mathcal{M})_{(2)}^T] - \zeta [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ (\sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] \end{array} \right\} U_2\} \end{array} \right\} \\
&= \arg\max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T (B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}) U_2\} - \{ \sum_{k=I_1'+1}^{I_1} tr\{U_2^T \\ \left\{ \begin{array}{l} [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT} \\ (\mathcal{M}_i - \mathcal{M})_{(2)}^T] - \zeta [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] \end{array} \right\} U_2\}\} \end{array} \right\} \\
&= \arg\max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T (B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}) U_2\} - \{ \sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT} \\ \left\{ \begin{array}{l} [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)}^T U_2 U_2^T \\ (\mathcal{M}_i - \mathcal{M})_{(2)}] - \\ \zeta [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ U_2 U_2^T (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}] \end{array} \right\} \mathbf{u}_{1,k}^t\}\} \end{array} \right\}
\end{aligned}
$$

$$= \arg\max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T (B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}) U_2\} - \{ \sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT} \\ \left\{ \begin{array}{l} [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)}^T \\ (\mathcal{M}_i - \mathcal{M})_{(2)}] - \\ \zeta [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T \\ (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}] \end{array} \right\} \mathbf{u}_{1,k}^t \}\} \end{array} \right\}$$

$$= \arg\max_{U_2^T U_2 = I} \left\{ \begin{array}{l} tr\{U_2^T (B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}) U_2\} - \\ \{ \sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT} (B_1^{\bar{1},\bar{2}} - \zeta W_1^{\bar{1},\bar{2}}) \mathbf{u}_{1,k}^t \}\} \end{array} \right\} \tag{3.8}$$

$$= \arg\max_{U_2^T U_2 = I} tr\{U_2^T (B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}) U_2\} \tag{3.9}$$

As we can see, the second term in (Equation 3.8) is a fixed value that cannot be changed at this step. Thus, optimizing (Equation 3.8) is equivalent to optimizing (Equation 3.9). Also, we know that the second term in (Equation 3.8) will achieve its minimum when U_1^t are obtained through SVD on $B_1^{\bar{1},\bar{2}} - \zeta W_1^{\bar{1},\bar{2}}$. In other words, if we obtain U_2 through SVD on $B_2^{\bar{1},\bar{2}} - \zeta W_2^{\bar{1},\bar{2}}$ and U_1 through SVD on $B_1^{\bar{1},\bar{2}} - \zeta W_1^{\bar{1},\bar{2}}$, the mode-2 objective function will achieve its global maximum value. Same thing holds for the mode-1 objective function. Thus, we are able to achieve the optimal projection of GTDA directly without any iteration. This completes the proof.

Based on the above theorem, we propose Direct General Tensor Discriminant Analysis (DGTDA) whose training stage is summarized in Table Table IV.

TABLE IV

TRAINING PROCEDURE OF DGTDA

Input:	Training tensors $\mathcal{X}_{i,j} \mid_{1\le i\le c}^{1\le j\le n_i} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, their class labels $i \in \{1, 2, \ldots, c\}$, dimensionality of the reduced tensor subspace $I'_1 \times I'_2 \times \ldots \times I'_N$.
Step 1.	For $n = 1, 2, \ldots, N$ do{
Step 2.	Calculate $B_n^{\bar{1},\bar{2},\ldots,\bar{N}} = \sum_{i=1}^{c} n_i[(\mathcal{M}_i - \mathcal{M})_{(n)}][(\mathcal{M}_i - \mathcal{M})_{(n)}]^T$;
Step 3.	Calculate $W_n^{\bar{1},\bar{2},\ldots,\bar{N}} = \sum_{i=1}^{c}\sum_{j=1}^{n_i}[(\mathcal{X}_{i,j} - \mathcal{M}_i)_{(n)}][(\mathcal{X}_{i,j} - \mathcal{M}_i)_{(n)}]^T$;
Step 4.	Optimize $U_n^\star = \arg\max_{U^TU=I} tr[U^T(B_n^{\bar{1},\bar{2},\ldots,\bar{N}} - \zeta W_n^{\bar{1},\bar{2},\ldots,\bar{N}})U]$ by SVD on $B_n^{\bar{1},\bar{2},\ldots,\bar{N}} - W_n^{\bar{1},\bar{2},\ldots,\bar{N}}$, where $\zeta = \lambda_{max}((W_n^{\bar{1},\bar{2},\ldots,\bar{N}})^{-1}B_n^{\bar{1},\bar{2},\ldots,\bar{N}})$. }End For loop in Step 1.
Step 5.	$\mathcal{Y}_{i,j} = \mathcal{X}_{i,j}\prod_{k=1}^{N}\times_k U_k^T$.
Output:	The projection matrices $U_n \in \mathbb{R}^{I_n\times I'_n} \mid_{n=1}^{N}$ constrained by $U_n^TU_n = I$ and the projected tensors $\mathcal{Y}_{i,j} \mid_{1\le i\le c}^{1\le j\le n_i} \in \mathbb{R}^{I'_1\times I'_2\times\ldots\times I'_N}$.

3.4.2 Constrained Multilinear Discriminant Analysis

Table Table V describes the training stage of DATER. Due to the fact that DATER does not converge over iterations, its performance is highly affected by the number of iterations executed upon termination. It is also impossible to determine when to terminate the algorithm. We next propose an iterative MDA method, Constrained Multilinear Discriminant Analysis (CMDA), whose training procedure is summarized in Table Table VI. The main difference of CMDA in comparison to DATER lies in step 5 where a new U_n at iteration t is sought under the constraint that $U_nU_n^T = I$. As in GTDA, such U_n can be obtained by SVD on $(W_n^{\bar{n}})^{-1}B_n^{\bar{n}}$. We demonstrate that unlike DATER, the value of the scatter ratio criterion in CMDA approaches its extreme value, if it exists, with bounded error in the limit. Such property leads to superior and stabler

TABLE V

TRAINING PROCEDURE OF DATER

Input:	Training tensors $\mathcal{X}_{i,j} \mid_{1\le i\le c}^{1\le j\le n_i} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, their class labels $i \in \{1, 2, ..., c\}$, dimensionality of the reduced tensor subspace $I'_1 \times I'_2 \times ... \times I'_N$, the maximum number of iterations T_{max}.
Initialization:	Initialize $U_n^0 = I_{I_n\times I'_n} \mid_{n=1}^N$.
Step 1.	For $t = 1, 2, ..., T_{max}\{$
Step 2.	For $n = 1, 2, ..., N$ do$\{$
Step 3.	Calculate $B_n^{\bar{n}t} = \sum_{i=1}^c n_i[(\mathcal{M}_i - \mathcal{M}) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}[(\mathcal{M}_i - \mathcal{M}) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}^T$;
Step 4.	Calculate $W_n^{\bar{n}t} = \sum_{i=1}^c \sum_{j=1}^{n_i}[(\mathcal{X}_{i,j} - \mathcal{M}_i) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}[(\mathcal{X}_{i,j} - \mathcal{M}_i) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}^T$;
Step 5.	Optimize $U_n^{\star t} = \arg\max \frac{tr(U^T B_n^{\bar{n}t} U)}{tr(U^T W_n^{\bar{n}t} U)}$ by eigenvalue decomposition on $(W_n^{\bar{n}t})^{-1} B_n^{\bar{n}t}$. $\}$End For loop in Step 2.
Step 6.	If $t > 2$ and $\|U_n^t - U_n^{t-1}\| < I'_n I_n \varepsilon, n = 1, \ldots, N$, break. $\}$End For loop in Step 1.
Output:	The projection matrices $U_n \in \mathbb{R}^{I_n\times I'_n} \mid_{n=1}^N$.

classification performance in comparison to DATER. To our best knowledge, CMDA is the first scatter ratio maximization-based MDA method that exhibits convergency.

We first define a new term "asymptotically bounded sequence" in order to describe the phenomenon we observe.

Asymptotically Bounded Error A sequence f_n has asymptotically bounded error if $e_n = u_n - l_n$ converges, i.e. $\lim_{n\to+\infty} e_n = c$ where $l_n \leq f_n \leq u_n$.

Asymptotically Bounded Sequence A sequence f_n is an asymptotically bounded sequence if f_n has asymptotically bounded error and its boundary sequences u_n and l_n converge where $l_n \leq f_n \leq u_n$.

If one of the bounds of f_n converges, say, $\lim_{n\to+\infty} u_n = u$, then $\lim_{n\to+\infty} l_n = u - c$. Hence in the limit, the value of f_n will be lower bounded by $u - c$ and upper bounded by u. In fact, there are two possible behaviors of f_n in the long run, one is that it actually converges to some limit value as its boundary sequences do, the other is that it always oscillates around the value $u - c/2$ and the variation of its value is bounded by $\frac{c}{2}$. Although it does not really converge in the latter case, since the variation is bounded, its behavior is to some extent stable.

Let S_n be the set that includes all possible U_n, that is, $U_n \in S_n$ constrained by $U_n^T U_n = I$, for $n = 1, 2, \ldots N$. Define a continuous function $f : S_1 \times S_2 \times \ldots \times S_N \to \mathbb{R}^+$:

$$f(U_n|_{n=1}^N) = \frac{\sum_{i=1}^{c} n_i \|(\mathcal{M}_i - \mathcal{M}) \prod_{k=1}^{N} \times_k U_k^T\|_F^2}{\sum_{i=1}^{c} \sum_{j=1}^{n_i} \|(\mathcal{X}_{i,j} - \mathcal{M}_i) \prod_{k=1}^{N} \times_k U_k^T\|_F^2}.$$

Then construct N different mappings based on f:

TABLE VI

TRAINING PROCEDURE OF CMDA

Input:	Training tensors $\mathcal{X}_{i,j} \mid_{1\le i\le c}^{1\le j\le n_i} \in \mathbb{R}^{I_1\times I_2\times\ldots\times I_N}$, their class labels $i \in \{1,2,...,c\}$, dimensionality of the reduced tensor subspace $I'_1 \times I'_2 \times ... \times I'_N$, the maximum number of iterations T_{max}.
Initialization:	Initialize $U_n^0 = 1_{I_n\times I'_n} \mid_{n=1}^{N}$ (All entries of U_n^0 are 1).
Step 1.	For $t = 1,2,...,T_{max}\{$
Step 2.	For $n = 1,2,...,N$ do$\{$
Step 3.	Calculate $B_n^{\bar{n}t} = \sum_{i=1}^{c} n_i[(\mathcal{M}_i - \mathcal{M}) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}[(\mathcal{M}_i - \mathcal{M}) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}^T$;
Step 4.	Calculate $W_n^{\bar{n}t} = \sum_{i=1}^{c} \sum_{j=1}^{n_i}[(\mathcal{X}_{i,j} - \mathcal{M}_i) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}[(\mathcal{X}_{i,j} - \mathcal{M}_i) \times_1 U_1^{tT} \times \ldots \times_{n-1} U_{n-1}^{tT} \times_{n+1} U_{n+1}^{t-1T} \times \ldots \times_N U_N^{t-1T}]_{(n)}^T$;
Step 5.	Optimize $U_n^{\star t} = \arg\max_{U^TU=I} \frac{tr(U^T B_n^{\bar{n}t} U)}{tr(U^T W_n^{\bar{n}t} U)}$ by SVD on $(W_n^{\bar{n}t})^{-1} B_n^{\bar{n}t}$. $\}$End For loop in Step 2.
Step 6.	Check convergence: $\sum_{n=1}^{N} \|U_n^t (U_n^{(t-1)})^T - I\| \le \varepsilon$. $\}$End For loop in Step 1.
Step 7.	$\mathcal{Y}_{i,j} = \mathcal{X}_{i,j} \prod_{k=1}^{N} \times_k U_k^T$.
Output:	The projection matrices $U_n \in \mathbb{R}^{I_n\times I'_n} \mid_{n=1}^{N}$ constrained by $U_n^T U_n = I$ and the projected tensors $\mathcal{Y}_{i,j} \mid_{1\le i\le c}^{1\le j\le n_i} \in \mathbb{R}^{I'_1\times I'_2\times\ldots\times I'_N}$.

$f_n(U_n) = f(U_n; U_k|_{k=1}^{n-1}, U_k|_{k=n+1}^{N})$, for $n = 1, 2, \ldots N$, where $f(U_n; U_k|_{k=1}^{n-1}, U_k|_{k=n+1}^{N})$ means that the function f varies with U_n with fixed $U_k|_{k=1}^{n-1}$ and $U_k|_{k=n+1}^{N}$. Based on these mappings, we define

$$g_n(U_n) = \arg \max_{U_n^T U_n = I, U_n \in S_n} f_n(U_n) = \arg \max_{U_n^T U_n = I} \frac{tr\{U_n^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(n)} (\otimes_{k=N, k\neq n}^{1} U_k U_k^T) (\mathcal{M}_i - \mathcal{M})_{(n)}^T] U_n\}}{tr\{U_n^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(n)} (\otimes_{k=N, k\neq n}^{1} U_k U_k^T) (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(n)}^T] U_n\}},$$

where the mapping $g_n(U_n)$ is calculated by arg maximizing $f_n(U_n^t)$ at the t^{th} iteration with the given $U_k^t|_{k=1}^{n-1}$ in the t^{th} iteration and $U_k^{t-1}|_{k=n+1}^{N}$ in the $t-1^{th}$ iteration.

Given randomly initialized $U_l^{(0)} \in S_l$ for $l = 1, 2, \ldots N$, the iterative optimization procedure of CMDA generates a sequence of items $U_l^{(t)}$ via $g_l(U_l)$ which correspond to a sequence of items $f_l(U_l^{(t)})$, called the objective function sequence.

Theorem 3.4.2 *Given randomly initialized $U_n^0 \in S_n$, $n = 1, 2, \ldots N$, the objective function sequence $f_l(U_l^{(t)})$ generated by CMDA iterative optimization procedure is an asymptotically bounded sequence.*

Proof For simplicity, we prove the 2^{nd} order case, where all samples are matrices$\in \mathbb{R}^{I_1 \times I_2}$ and we aim to find two projection matrices $U_1 \in \mathbb{R}^{I_1 \times I_1'}$ and $U_2 \in \mathbb{R}^{I_2 \times I_2'}$. This proof can be easily generalized to higher-order case.

Suppose at iteration t, we obtained U_1^t by SVD on some matrix and keep only the first I_1' columns so that $U_1^t \in \mathbb{R}^{I_1 \times I_1'}$. Therefore, $U_1^t U_1^{tT} = \sum_{k=1}^{I_1'} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT} = I - \sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}$. Consequently, at iteration t, the optimal U_2^t can be obtained by

$$
\begin{aligned}
U_2^{\star t} &= \arg \max_{U_2^T U_2 = I} \frac{tr(U_2^T B_2 U_2)}{tr(U_2^T W_2 U_2)} \\
&= \arg \max_{U_2^T U_2 = I} \frac{tr\left\{ \begin{array}{l} U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} U_1^t \\ U_1^{tT} (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2 \end{array} \right\}}{tr\left\{ \begin{array}{l} U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} U_1^t \\ U_1^{tT} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2 \} \end{array} \right\}} \\
&= \arg \max_{U_2^T U_2 = I} \frac{\left\{ \begin{array}{l} tr\{ U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} \\ (I - \sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2 \} \end{array} \right\}}{\left\{ \begin{array}{l} tr\{ U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} \\ (I - \sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2 \} \end{array} \right\}} \\
&= \arg \max_{U_2^T U_2 = I} \frac{\left\{ \begin{array}{l} tr\{ U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2 \} - \\ tr\{ U_2^T [\sum_{i=1}^{c} n_i (\mathcal{M}_i - \mathcal{M})_{(2)} (\sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) \\ (\mathcal{M}_i - \mathcal{M})_{(2)}^T] U_2 \} \end{array} \right\}}{\left\{ \begin{array}{l} tr\{ U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2 \} \\ -tr\{ U_2^T [\sum_{i=1}^{c} \sum_{j=1}^{n_i} (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)} (\sum_{k=I_1'+1}^{I_1} \mathbf{u}_{1,k}^t \mathbf{u}_{1,k}^{tT}) \\ (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] U_2 \} \end{array} \right\}}
\end{aligned}
$$

$$
\begin{aligned}
&= \arg\max_{U_2^T U_2 = I} \frac{\left\{ \begin{array}{l} tr\{U_2^T[\sum_{i=1}^{c} n_i(\mathcal{M}_i - \mathcal{M})_{(2)}(\mathcal{M}_i - \mathcal{M})_{(2)}^T]U_2\} - \\ \{\sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT}[\sum_{i=1}^{c} n_i(\mathcal{M}_i - \mathcal{M})_{(1)} \\ (\mathcal{M}_i - \mathcal{M})_{(1)}^T]\mathbf{u}_{1,k}^t\}\} \end{array} \right\}}{\left\{ \begin{array}{l} tr\{U_2^T[\sum_{i=1}^{c}\sum_{j=1}^{n_i}(\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}(\mathcal{X}_{i,j} - \mathcal{M}_i)_{(2)}^T] \\ U_2\} - \{\sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT}[\sum_{i=1}^{c}\sum_{j=1}^{n_i}(\mathcal{X}_{i,j} - \mathcal{M}_i)_{(1)}) \\ (\mathcal{X}_{i,j} - \mathcal{M}_i)_{(1)}^T]\mathbf{u}_{1,k}^t\}\} \end{array} \right\}} \\
&= \arg\max_{U_2^T U_2 = I} \left\{ \frac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\} - \left\{ \sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t\} \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\} - \left\{ \sum_{k=I_1'+1}^{I_1} tr\{\mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t\} \right\}} \right\} \\
&= \arg\max_{U_2^T U_2 = I} \left\{ \frac{\frac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}}}{\frac{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}}} \right\} \\
&= \arg\max_{U_2^T U_2 = I} \left\{ \frac{\frac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}}}{1 - \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}} \right\}
\end{aligned}
$$

$$= \arg\max_{U_2^T U_2 = I} \frac{\frac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \left\{ \frac{\sum_{k=1}^{I_1'} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} \right\}^{-1}}{1 - \left\{ \frac{\sum_{k=1}^{I_1'} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + 1 \right\}^{-1}}$$

$$= \arg\max_{U_2^T U_2 = I} \frac{\frac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \left\{ \frac{\sum_{k=1}^{I_1'} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} \right\}^{-1}}{1 - \left\{ \frac{\sum_{k=1}^{I_1'} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + \frac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + 1 \right\}^{-1}}$$

$$= \arg\max_{U_2^T U_2 = I} \frac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}}.$$

Denote the maximum value and the minimum value of the term $\frac{\sum_{k=1}^{I_1'} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}$ by a_{max} and a_{min} respectively. Then the objective function $f_2(U_2^t)$ is upper bounded by function

$$f_2(U_2^t)^{upper} = \frac{\dfrac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \left\{ \begin{array}{l} a_{max} + \\ \dfrac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} \end{array} \right\}^{-1}}{1 - \left\{ a_{min} \dfrac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + 1 \right\}^{-1}}$$

and lower bounded by function

$$f_2(U_2^t)^{lower} = \frac{\dfrac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}} - \left\{ \begin{array}{l} a_{min} + \\ \dfrac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} \end{array} \right\}^{-1}}{1 - \left\{ a_{max} \dfrac{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} B_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}}{\sum_{k=I_1'+1}^{I_1} tr\left\{ \mathbf{u}_{1,k}^{tT} W_1^{\bar{1},\bar{2}} \mathbf{u}_{1,k}^t \right\}} + 1 \right\}^{-1}}$$

As the CMDA iterative procedure proceeds, both the values of $f_2(U_2^t)^{upper}$ and $f_2(U_2^t)^{lower}$ increase monotonically. Moreover, they both achieve their maximum values, denoted by f^{upper} and f^{lower} respectively, when $U_2 = \arg \max_{U_2^T U_2 = I} \dfrac{tr\left\{ U_2^T B_2^{\bar{1},\bar{2}} U_2 \right\}}{tr\left\{ U_2^T W_2^{\bar{1},\bar{2}} U_2 \right\}}$ and $U_1 = \arg \max_{U_1^T U_1 = I} \dfrac{tr\left\{ U_1^T B_1^{\bar{1},\bar{2}} U_1 \right\}}{tr\left\{ U_1^T W_1^{\bar{1},\bar{2}} U_1 \right\}}$. Hence, both $f_2(U_2^t)^{upper}$ and $f_2(U_2^t)^{lower}$ converge monotonically. Similarly, we can prove that $f_1(U_1^t)$ is upper and lower bounded by two monotonically convergent sequences $f_1(U_1^t)^{upper}$ and

$f_1(U_1^t)^{lower}$. Since $f_2(U_2^t)^{upper} > f_1(U_1^t)^{upper}$ and $f_2(U_2^t)^{lower} > f_1(U_1^t)^{lower}$, together we have the following relationship:

$$\begin{aligned} &f_1(U_1^t)^{upper} < f_2(U_2^t)^{upper} < f_1(U_1^{t+1})^{upper} < f_2(U_2^{t+1})^{upper} \ldots \leq f^{upper}; \\ &f_1(U_1^t)^{lower} < f_2(U_2^t)^{lower} < f_1(U_1^{t+1})^{lower} < f_2(U_2^{t+1})^{lower} \ldots \leq f^{lower}. \end{aligned} \tag{3.10}$$

Therefore, the sequence $f_l(U_l^{(t)})$ generated by the CMDA iterative optimization procedure is upper and lower bounded by two monotonically convergent sequences, thus it is an asymptotically bounded sequence. This completes the proof.

As a result, we can stop the iterative optimization procedure when the change of f between two successive iterations is small enough. To be more specific, the algorithm either halts when f reaches its extreme value or the change of f between two successive iterations is always smaller than the threshold $\varepsilon = f^{upper} - f^{lower}$.

3.4.3 Algorithm Analysis and Discussions

To show that $W_n^{\bar{n}}$ is generally nonsingular in CMDA and DGTDA, we only need to compare the original feature space dimensionality to $n-c$ where n is the sample number and c is the class number as in Section 3.2.2. In traditional LDA, a sample must be vectorized, in our case, the dimensionality of the sample after vectorization is $I_1 \times I_2 \times \ldots \times I_N$, which is a considerably large number compared to $n - c$. However in CMDA and DGTDA, the sample is now represented in tensor form and we deal with each tensor mode separately. The original feature space dimensionality of each mode is I_n, which is much smaller than $I_1 \times I_2 \times \ldots \times I_N$. Hence it is less

possible to result in singular S_W. In conclusion, the proposed methods overcome the frequently appeared USP in LDA by employing tensor representation and conducting optimization on each tensor mode separately.

For ease of understanding, let us assume that the sample tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ has uniform dimensionality in all modes, that is, $I_1 = I_2 = \ldots = I_N = L$. Then Table Table VII compares the time and space complexities of all the algorithms discussed above. Here T denotes the number of iterations.

TABLE VII

COMPUTATIONAL COMPLEXITY ANALYSIS

Method	Time Complexity	Space Complexity
LDA	$O(L^{3N})$	$O(L^{2N})$
GTDA/DATER/CMDA	$O(TNL^3)$	$O(NL^2)$
DGTDA	$O(NL^3)$	$O(NL^2)$

3.5 Multilinear Discriminant Analysis: Classification

By the end of the training stage, with the learned projection matrices $U_n|_{n=1}^N$, the lower-dimensional tensor subspace representation $\mathcal{Y}_{i,j}$ of each $\mathcal{X}_{i,j}$ belonging to class i is computed as in (Equation 3.4). When a query tensor $\mathcal{X}$ is received, we first compute its tensor subspace representation by

$\mathcal{Y} = \mathcal{X} \prod_{k=1}^N \times_k U_k^T \in \mathbb{R}^{I_1' \times I_2' \times \ldots \times I_N'}$.

We then compare the Frobenius distance between $\mathcal{Y}$ and each $\mathcal{Y}_{i,j}$, the class label of X is determined by the label of the training tensor that has the minimum distance,

$i^* = \arg\min_i \|\mathcal{Y}_{i,j} - \mathcal{Y}\|_F$.

In the following experiment section, we use this classification method for all algorithms for its simplicity in computation.

3.6 Experimental Results

To evaluate the effectiveness of our proposed methods, we compare DGTDA and CMDA with two other MDA methods that employ TTP, DATER and GTDA for face image ensemble recognition on the extended Yale Face Database B (29). Classifier introduced in Section 3.5 is employed for all algorithms. All the algorithms are implemented and run on a desktop with 8G RAM and Intel Core i7 CPU without code optimization.

3.6.1 The Extended Yale Face Database B and Experiment Settings

The extended Yale Face Database B contains single light source images of 28 human subjects under 9 poses and 64 illumination conditions. To speedup calculations, we crop the image to keep only the center area that contains face and resize each cropped face image to have 73×55 pixels. For each subject, poses 0, 2, 4, 6, 8 are used to form 5 3rd-order tensors as training samples for that class, each of size $73 \times 55 \times 64$. In other words, each sample tensor contains 64 images under various illumination conditions with one pose of the same person. Similarly, poses 1, 3, 5, 7 form 4 test tensors of each class. To sum up, the number of classes c is 28. Within each class, the number of training tensor samples n_i is 5 for $i = 1, 2, \ldots c$. The number of testing query tensors for each class is 4, so in total, there are $4 \times 28 = 112$ tensor samples that are used

to test the classification performance. 64 images of a single person under various illumination conditions with one pose, which form a sample tensor, are shown in Figure Figure 6. Note that there are 18 images corrupted during acquisition, but we use them in the training process as well.

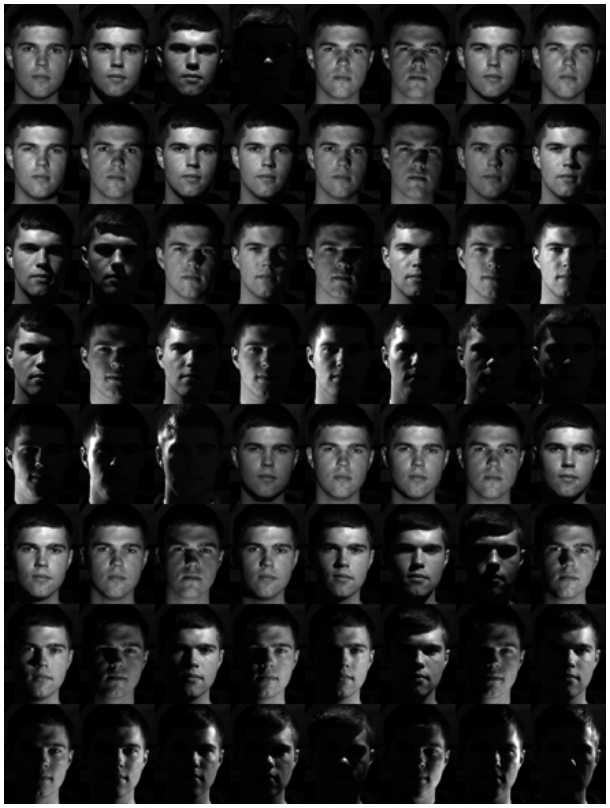


Figure 6. 64 images from a sample tensor.

There are two parameters that need to be determined. The first one is the user-controlled tuning parameter ζ in GTDA. As illustrated in (19), in real-world applications, a manually chosen value of ζ always achieves better prediction results than the calculated value, we set ζ to be the largest eigenvalue of $(W_n^{\bar{n}})^{-1}B_n^{\bar{n}}$ for each mode objective function in GTDA. Note that it varies as iteration continues. However, in DGTDA, ζ is determined. The other parameter is the threshold ε used to check convergence of the iterative optimization procedures in GTDA and CMDA. Denote the mode-n convergence error at iteration t by $Err_n^t = \|U_n^t(U_n^{t-1})^T - I\|_F$ for $n = 1, 2, 3$. Thus the total convergence error $Err^t = \sum_{i=1}^{3} Err_n^t$. Then we say the algorithm converges when $|Err_n^t - Err_n^{t-1}| \leq \varepsilon$ for $n = 1, 2, 3$, where $\varepsilon = 10^{-5}$. The maximum number of iteration T_{max} is chosen to be 50. Since DATER does not converge, we choose its number of iterations to be the same as the number of iterations CMDA takes to converge.

3.6.2 Performance Evaluation

For simplicity, we assume that all tensor modes have the same reduced dimensionality, that is, $I_1' = I_2' = I_3' = Dim$. We vary Dim from 1 to 54 to test the performance of each method with different reduced dimensionality, the results are shown in Figure Figure 7. To show that there does not exist a direct solution to CMDA, we also compare its performance with "a direct solution", denoted by DCMDA where the projection matrix U_n for mode n is obtained independently with other matrices being set to I, like in DGTDA. In general, MDA methods based on scatter ratio maximization (CMDA, DATER, even DCMDA) outperform those based on scatter difference maximization (DGTDA, GTDA). To be more specific, we make the following observations: 1) the fact that CMDA always performs better than DCMDA

confirms that DCMDA is not the optimal solution for CMDA, hence there does not exist a direct solution to CMDA as DGTDA to GTDA; 2) when using the same number of iterations, DATER achieves higher accuracy for lower-dimensionality 1 to 23 while CMDA performs better between dimensionality 24 and 54; in fact, the performance of DATER vary dramatically as the reduced dimensionality *Dim* increases; 3) the performance of DGTDA is as good as, if not better, that of GTDA at a majority of reduced dimensionality.

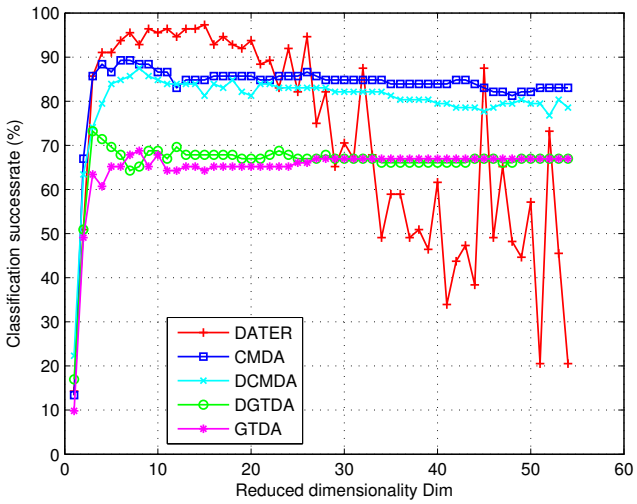


Figure 7. Classification accuracy comparison.

In terms of training time costs, experimental results coincide with theoretical predictions that the time cost for DGTDA is the same as the average time cost for one iteration of GTDA and the average time cost for each iteration of CMDA is almost the same as DATER. Therefore, we summarize in Table Table VIII the time cost for each iteration of GTDA and CMDA during training procedure.

3.6.3 Convergence Examination

Table Table IX summarizes the number of iterations it takes for GTDA and CMDA to converge, as well as the Err values upon termination of iteration. GTDA usually takes 3 to 6 iterations to converge while CMDA takes 30 on average, if it converges. There are only 10 out of 54 cases in total that CMDA does not converge within 50 number of iterations. This indicates that although in Theorem 3.4.2 we only prove that the objective function sequence of CMDA is asymptotically bounded, it could actually be convergent.

We specifically look into the convergence of CMDA and GTDA when $Dim = 4$ and $Dim = 44$ for example. Figure 8(a) and Figure 8(b) compare the convergence error Err_1, Err_2, Err_3 and Err with respect to the number of iteration t when $Dim = 4$. Similarly, Figure 8(c) and Figure 8(d) show comparison results when $Dim = 44$. In both cases, the two methods converge over iterations. According to the proof of Theorem 3.4.2, CMDA converges to its optimum, if it exists, with oscillations, hence it takes more iterations than GTDA to converge.

Also, we examine the classification performance of CMDA, GTDA and DATER with respect to the number of iteration. Figures 9(a) and 9(b) display the results. Due to the convergence of CMDA and GTDA, they both provide relatively stable classification performance as they

approach convergence. However, the performance of DATER varies dramatically as iteration continues and it is hard to determine when to terminate the iterative procedure so as to achieve good performance.

3.7 CONCLUSION

Linear discriminant analysis (LDA) relies on data representation in the form of vectors. However, real-world data are usually generated from the interaction of multiple factors, and thus naturally employ higher-order tensor representations. As a result, recent efforts have been made to extend LDA for tensor data classification, which is generally referred to as the multilinear discriminant analysis (MDA) problem. In this paper, we propose two MDA methods, DGTDA and CMDA, which seek an optimal tensor-to-tensor projection (TTP) for discrimination in a lower-dimensional tensor subspace. DGTDA finds the projection directly by maximizing the scatter difference criterion while CMDA learns the projection iteratively by maximizing the scatter ratio criterion. We demonstrate that DGTDA outperforms GTDA in terms of both training efficiency and classification accuracy. In addition, we prove theoretically that in the limit, the value of the scatter ratio criterion in CMDA approaches its extreme value, if it exists, with bounded error. In fact, experimental results show that in most cases, the optimization procedure of CMDA converges, thus leading to superior and stabler classification performance in comparison to DATER. To our best knowledge, CMDA is the first scatter ratio maximization-based MDA method that exhibits convergency.

Finally, there are still some aspects of the proposed methods that deserve further study. We only prove that the objective function sequence generated by CMDA iterative procedure

is asymptotically bounded. We are not sure if such procedure always converges. If not, what factors affect its convergency? Besides, both DGTDA and CMDA are multilinear, they can not capture the nonlinear structure of the data manifold effectively. It remains unclear how to generalize our approach to nonlinear case.

TABLE VIII

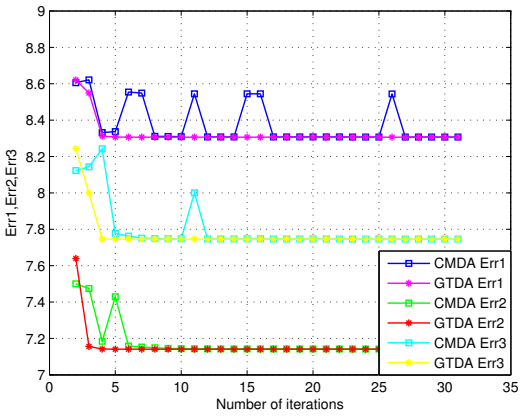
TRAINING TIME COMPARISON (SECONDS)

	avg. time/iteration			avg. time/iteration	
Dim	GTDA	CMDA	Dim	GTDA	CMDA
1	27.2	19.5	28	29.0	28.6
2	27.3	26.2	29	29.1	28.7
3	27.1	26.2	30	29.0	28.5
4	27.2	26.9	31	29.0	28.5
5	27.2	26.9	32	29.3	28.6
6	27.1	27.0	33	29.2	28.7
7	27.1	27.0	34	29.7	28.8
8	27.4	27.0	35	30.0	29.3
9	27.4	27.1	36	30.0	29.1
10	27.4	27.1	37	30.2	28.9
11	27.4	27.0	38	30.5	29.2
12	27.5	27.1	39	30.5	29.4
13	27.6	27.2	40	30.8	29.3
14	27.4	27.2	41	31.0	29.5
15	27.4	27.6	42	31.3	29.5
16	27.6	27.3	43	31.5	29.6
17	27.8	27.5	44	31.7	29.7
18	27.8	27.4	45	32.2	29.8
19	28.0	27.6	46	32.2	29.9
20	27.8	27.7	47	32.3	30.2
21	28.0	27.7	48	32.4	30.0
22	28.0	27.6	49	32.8	30.3
23	28.0	27.8	50	33.0	30.5
24	28.0	27.8	51	34.5	30.6
25	28.3	27.9	52	35.3	30.8
26	28.1	28.0	53	35.2	30.8
27	28.5	28.1	54	35.8	30.9

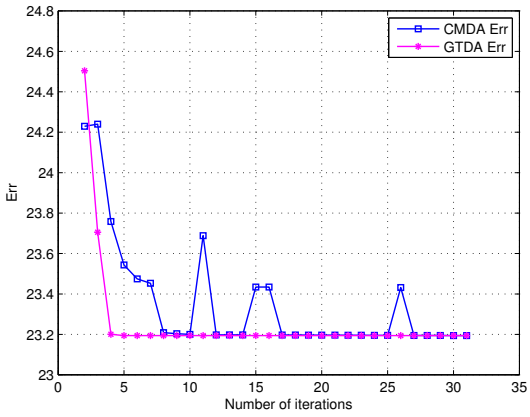
TABLE IX

CONVERGENCE EXAMINATION

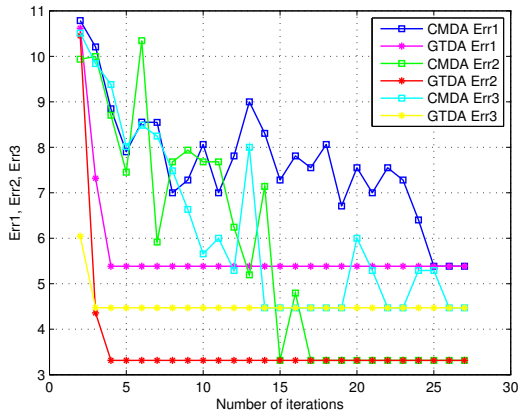
	♯ of iterations		Err			♯ of iterations		Err	
Dim	GTDA	CMDA	GTDA	CMDA	Dim	GTDA	CMDA	GTDA	CMDA
1	3	5	23.7710	23.7710	28	4	28	17.9044	17.9044
2	4	≥ 50	23.5803	23.5806	29	4	20	17.6483	17.6484
3	5	≥ 50	23.3880	23.3890	30	4	15	17.3884	17.3884
4	5	31	23.1940	23.1945	31	4	16	17.1243	17.1243
5	6	≥ 50	22.9984	22.9992	32	4	20	16.8558	16.8558
6	4	≥ 50	22.8011	22.8012	33	6	22	16.5827	16.5827
7	4	26	22.6021	22.6023	34	5	27	16.3048	16.3048
8	3	43	22.4012	22.4013	35	4	12	16.0217	16.0217
9	3	39	22.1985	22.1990	36	4	16	15.7332	15.7332
10	3	≥ 50	21.9939	22.0014	37	4	14	15.4388	15.4388
11	3	21	21.7874	21.7881	38	4	17	15.1382	15.1382
12	4	≥ 50	21.5788	21.8319	39	4	16	14.8310	14.8310
13	3	≥ 50	21.3681	21.3685	40	4	13	14.5165	14.5165
14	3	≥ 50	21.1553	21.4341	41	4	13	14.1943	14.1944
15	3	≥ 50	20.9403	22.0450	42	4	17	13.8637	13.8637
16	3	28	20.7230	20.7231	43	4	13	13.5239	13.5239
17	3	26	20.5034	20.5034	44	4	11	13.1739	13.1739
18	3	21	20.2813	20.2813	45	4	15	12.8127	12.8127
19	4	16	20.0567	20.0567	46	4	13	12.4388	12.4388
20	3	16	19.8294	19.8294	47	4	20	12.0506	12.0506
21	5	16	19.5995	19.5995	48	5	11	11.6458	11.6458
22	4	26	19.3667	19.3668	49	4	10	11.2215	11.2215
23	4	22	19.1310	19.1311	50	4	11	10.7736	10.7736
24	5	21	18.8923	18.8924	51	4	≥ 50	10.2960	11.5608
25	4	23	18.6504	18.6504	52	4	11	9.7787	9.7787
26	4	14	18.4052	18.4052	53	4	30	9.2030	9.2032
27	4	25	18.1566	18.1566	54	4	25	8.5212	8.5212



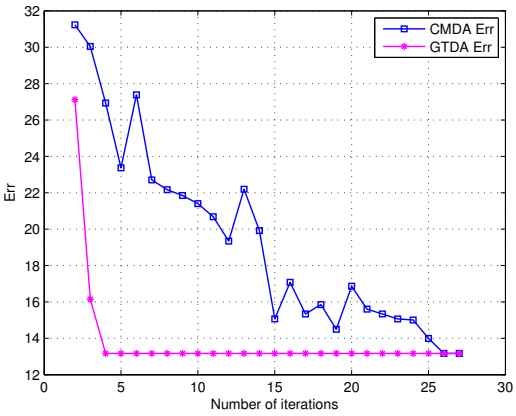
(a) Dim=4 Err1, Err2, Err3 vs. number of iteration



(b) Dim=4 Err vs. number of iteration

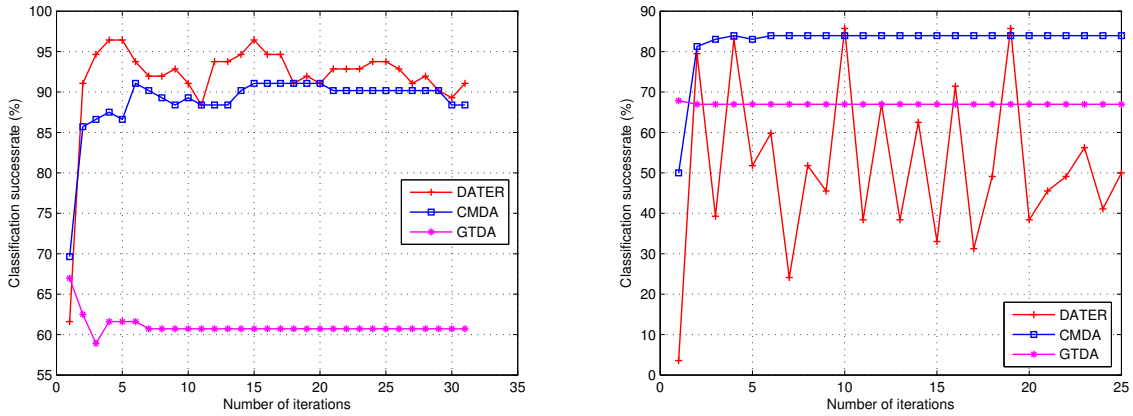


(c) Dim=44 Err1, Err2, Err3 vs. number of iteration



(d) Dim=44 Err vs. number of iteration

Figure 8. Convergence error examination.



(a) Dim=4 classification success rate vs. number of iteration

(b) Dim=44 classification success rate vs. number of iteration

Figure 9. Classification accuracy vs. number of iterations.

CHAPTER 4

GENERALIZED TENSOR COMPRESSIVE SENSING

Compressive sensing (CS) has triggered enormous research activity since its first appearance. CS exploits the signal's sparseness or compressibility in a particular domain and integrates data compression and acquisition. While conventional CS theory relies on data representation in the form of vectors, many data types in various applications such as color imaging, video sequences, and multi-sensor networks, are intrinsically represented by higher-order tensors. Application of CS to higher-order data representation is typically performed by conversion of the data to very long vectors that must be measured using very large sampling matrices, thus imposing a huge computational and memory burden. We propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order tensors. GTCS offers an efficient means for representation of multidimensional data by providing simultaneous acquisition and compression from all tensor modes. In addition, we compare the performance of the proposed method with Kronecker compressive sensing (KCS). We demonstrate experimentally that GTCS outperforms KCS in terms of both accuracy and speed.

4.1 Introduction

Recent literature has witnessed an explosion of interest in sensing that exploits structured prior knowledge in the general form of sparsity, meaning that signals can be represented by only a few coefficients in some domain. Central to much of this recent work is the paradigm of

compressive sensing (CS), also known under the terminology of compressed sensing, compressive sampling or compress sensing (30; 31; 32). CS theory permits relatively few linear measurements of the signal while still allowing exact reconstruction via nonlinear recovery process. The key idea is that the sparsity helps in isolating the original vector. The first intuitive approach to a reconstruction algorithm consists in searching for the sparsest vector that is consistent with the linear measurements. However, this ℓ_0-minimization problem is NP-hard in general and thus computationally infeasible. There are essentially two approaches for tractable alternative algorithms. The first is convex relaxation, leading to ℓ_1-minimization (33), also known as basis pursuit (34), whereas the second constructs greedy algorithms. Besides, in image processing, the use of total-variation minimization which is closely connected to ℓ_1-minimization first appears in (35) and is widely applied later on. By now basic properties of the measurement matrix which ensure sparse recovery by ℓ_1-minimization are known: the null space property (NSP) (36) and the restricted isometry property (RIP) (37).

An intrinsic limitation in conventional CS theory is that it relies on data representation in the form of vector. In fact, many data types do not lend themselves to vector data representation. For example, images are intrinsically matrices. As a result, great efforts have been made to extend traditional CS to CS of data in matrix representation. A straightforward implementation of CS on 2D images recasts the 2D problem as traditional 1D CS problem by converting images to long vectors, such as in (38). However, despite of considerably huge memory and computational burden imposed by long vector data and large sampling matrix, the sparse solutions produced by straightforward ℓ_1-minimization often incur visually unpleasant, high-frequency

oscillations. This is due to the neglect of attributes known to be widely possessed by images, such as smoothness. In (39), instead of seeking sparsity in the transformed domain, they proposed a total variation-based minimization to promote smoothness of the reconstructed image. Later, as an alternative for alleviating the huge computational and memory burden associated with image vectorization, block-based CS (BCS) was proposed in (40). In BCS, an image is divided into non-overlapping blocks and acquired using an appropriately-sized measurement matrix.

Another direction in the extension of CS to matrix CS generalizes CS concept and outlines a dictionary relating concepts from cardinality minimization to those of rank minimization (41; 42; 43). The affine rank minimization problem consists of finding a matrix of minimum rank that satisfies a given set of linear equality constraints. It encompasses commonly seen low-rank matrix completion problem (43) and low-rank matrix approximation problem as special cases. (41) first introduced recovery of the minimum-rank matrix via nuclear norm minimization. (42) generalized the RIP in (37) to matrix case and established the theoretical condition under which the nuclear norm heuristic can be guaranteed to produce the minimum-rank solution.

Real-world signals of practical interest such as color imaging, video sequences and multi-sensor networks, are usually generated by the interaction of multiple factors or multimedia and thus can be intrinsically represented by higher-order tensors. Therefore, the higher-order extension of CS theory for multidimensional data has become an emerging topic. One direction attempts to find the best rank-R tensor approximation as a recovery of the original data tensor as in (44), they also proved the existence and uniqueness of the best rank-r tensor approximation

in the case of 3rd order tensors. The other direction (45; 46) uses Kronecker product matrices in CS to act as sparsifying bases that jointly model the structure present in all of the signal dimensions as well as to represent the measurement protocols used in distributed settings. We propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order tensors. In addition, we propose two reconstruction procedures, a serial method (GTCS-S) and a parallelizable method (GTCS-P). Experimental results demonstrate the outstanding performance of GTCS in terms of both recovery accuracy and speed.

4.2 Background

4.2.1 Multilinear algebra

Outer Product and Tensor Product In linear algebra, the outer product typically refers to the tensor product of two vectors. $\mathbf{u} \circ \mathbf{v} = \mathbf{u}\mathbf{v}^\top$. In this paper, we won't differentiate between outer product and tensor product. To distinguish from Kronecker product, we use $\circ$ to denote tensor product of two vectors while $\otimes$ to denote Kronecker product. They can be related by $\mathbf{u} \circ \mathbf{v} = \mathbf{u} \otimes \mathbf{v}^\top$ and $vec(\mathbf{u} \circ \mathbf{v}) = \mathbf{v} \otimes \mathbf{u}$, where $vec(\cdot)$ denotes the vectorization of matrix along columns.

CANDECOMP/PARAFAC Decomposition (47)For a tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_d}$, the CANDECOMP/PARAFAC (CP) decomposition is $\mathcal{X} \approx [\lambda; A^{(1)}, A^{(2)}, ..., A^{(d)}] \equiv \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ ... \circ \mathbf{a}_r^{(d)}$, where $\lambda = [\lambda_1\ \lambda_2 ... \lambda_R]^\top \in \mathbb{R}^R$ and $A^{(i)} = [\mathbf{a}_1^{(i)}\ \mathbf{a}_2^{(i)} ... \mathbf{a}_R^{(i)}] \in \mathbb{R}^{N_i \times R}$ for $i = 1, ..., d$.

Core Tucker Decomposition Let $\mathcal{X} \in \mathbb{R}^{N_1 \times \ldots \times N_d}$ with mode-i unfolding $X_{(i)} \in \mathbb{R}^{N_i \times (N_1 \cdot \ldots \cdot N_{i-1} \cdot N_{i+1} \cdot \ldots \cdot N_d)}$. Denote by $R_i(\mathcal{X}) \subset \mathbb{R}^{N_i}$ the column space of $X_{(i)}$ whose rank is r_i. Let $\mathbf{c}_{1,i}, \ldots, \mathbf{c}_{r_i,i}$ be a basis

in $R_i(\mathcal{X})$. Then the subspace $\mathbf{V}(\mathcal{X}) := R_1(\mathcal{X}) \circ \ldots \circ R_d(\mathcal{X}) \subset \mathbb{R}^{N_1 \times \ldots \times N_d}$ contains $\mathcal{X}$. Clearly a basis in $\mathbf{V}$ consists of the vectors $\mathbf{c}_{i_1,1} \circ \ldots \circ \mathbf{c}_{i_d,d}$ where $i_j \in [r_j] := \{1, \ldots, r_j\}$ and $j \in [d]$. Hence the core Tucker decomposition of $\mathcal{X}$ is

$$\mathcal{X} = \sum_{i_j \in [r_j], j \in [d]} \xi_{i_1,\ldots,i_d} \mathbf{c}_{i_1,1} \circ \ldots \circ \mathbf{c}_{i_d,d}. \tag{4.1}$$

There are many ways to get a weaker decomposition as

$$\mathcal{X} = \sum_{i=1}^{K} \mathbf{a}_i^{(1)} \circ \ldots \circ \mathbf{a}_i^{(d)}, \quad \mathbf{a}_i^{(j)} \in R_j(\mathcal{X}), j \in [d]. \tag{4.2}$$

A simple constructive way is as follows. First decompose $X_{(1)}$ as $X_{(1)} = \sum_{j=1}^{r_1} \mathbf{c}_{j,1} \mathbf{g}_{j,1}^\top$ (e.g. by singular value decomposition (SVD)). Now each $\mathbf{g}_{j,1}$ can be viewed as a tensor of order $d-1$ $\in R_2(\mathcal{X}) \circ \ldots \circ R_d(\mathcal{X}) \subset \mathbb{R}^{N_2 \times \ldots \times N_d}$. Unfold each $\mathbf{g}_{j,1}$ in mode 2 to obtain $\mathbf{g}_{j,1_{(2)}}$ and decompose $\mathbf{g}_{j,1_{(2)}} = \sum_{l=1}^{r_2} \mathbf{d}_{l,2,j} \mathbf{f}_{l,2,j}^\top, \quad \mathbf{d}_{l,2,j} \in R_2(\mathcal{X}), \mathbf{f}_{l,2,j} \in R_3(\mathcal{X}) \circ \ldots \circ R_d(\mathcal{X})$. Continuing in this way we get a decomposition of type (Equation 4.2). Note that if $\mathcal{X}$ is s-sparse then each vector in $R_i(\mathcal{X})$ is s-sparse and each rank r_i is at most s. So $K \leq s^{d-1}$.

4.2.2 Compressive sensing

Compressive sensing is one of the ways to encode sparse information. A vector $\mathbf{x} \in \mathbb{R}^N$ is called s-sparse if it has at most s nonzero coordinates. The CS measurement protocol measures the signal $\mathbf{x}$ with the measurement matrix $A \in \mathbb{R}^{m \times N}$ where $m < N$ and transmits the encoded information $\mathbf{y} \in \mathbb{R}^m$ where $\mathbf{y} = A\mathbf{x}$. The receiver knows A and attempts to recover $\mathbf{x}$ from $\mathbf{y}$. Since $m < N$, there are usually infinitely many solutions for such under-constrained problem.

However, if $\mathbf{x}$ is known to be sufficiently sparse, then exact recovery of $\mathbf{x}$ is possible, which establishes the fundamental tenet of CS theory. The recovery is done by finding a solution $\mathbf{z}^\star \in \mathbb{R}^N$ satisfying

$$\mathbf{z}^\star = \arg\min\{\|\mathbf{z}\|_1,\ A\mathbf{z} = \mathbf{y}\}. \tag{4.3}$$

Such $\mathbf{z}^\star$ coincides with $\mathbf{x}$. The following well known result states that each s-sparse solution can be recovered uniquely if A satisfies the null space property of order s, denoted as NSP_s. That is, if $A\mathbf{w} = \mathbf{0}, \mathbf{w} \in \mathbb{R}^N \setminus \{\mathbf{0}\}$, then for any subset $S \subset \{1, \ldots, N\}$ with cardinality $|S| = s$ it holds that $\|\mathbf{v}_S\|_1 < \|\mathbf{v}_{S^c}\|_1$, where $\mathbf{v}_S$ denotes the vector that coincides with $\mathbf{v}$ on the index set S and is set to zero on S^c.

A simple way to generate such A is to use A sampled randomly from Gaussian or Bernoulli distributions. Then there exists a universal constant c such that if

$$m \geq 2cs \ln \frac{N}{s} \tag{4.4}$$

then the recovery of $\mathbf{x}$ using (Equation 4.3) is successful with probability at least $1 - \exp(-\frac{m}{2c})$.

Recently, the extension of CS theory for multidimensional signals has become an emerging topic. The objective of our paper is to consider the case where the s-sparse vector $\mathbf{x}$ is represented as an s-sparse tensor $\mathcal{X} = [x_{i_1,\ldots,i_N}] \in \mathbb{R}^{N_1 \times \ldots \times N_d}$. If we ignore the structure of $\mathcal{X}$ as a

tensor, and view it as a vector of size $N = N_1 \cdot \ldots \cdot N_d$, clearly, we can transmit $\mathcal{X}$ as $\mathbf{x}$ by using $\mathbf{y} = A\mathbf{x}$. If we use a random A as described above, we need m to be at least of order

$$m \geq 2cs(-\ln s + \sum_{i=1}^{d} \ln N_i). \tag{4.5}$$

In (46), Kronecker compressive sensing (KCS) constructs A from Kronecker product $A := U_1 \otimes \ldots \otimes U_d$, where $U_i \in \mathbb{R}^{m_i \times N_i}$ for $i = 1, \ldots, d$ and each U_i has NSP_s property. Then $\mathbf{x}$ is recovered uniquely from $\mathbf{y} = A\mathbf{x}$ by ℓ_1-minimization. In this paper, we analyze the compression and reconstruction of tensor $\mathcal{X}$ from the tensor $\mathcal{Y} = \mathcal{X} \times_1 U_1 \times \ldots \times_d U_d \in \mathbb{R}^{m_1 \times \ldots \times m_d}$ using a sequence of ℓ_1-minimizations similar to the minimization in (Equation 4.3). The advantage of our method is that our recovery problems are in terms of each U_i, which are much smaller comparing with the recovery related to A as given by the minimization in (Equation 4.3). This means that the amount of computations of our method is much less than that given by (Equation 4.3). If we choose our matrices U_i at random then we have the condition

$$m_i \geq 2cs \ln \frac{N_i}{s}, \quad i = 1, \ldots, d. \tag{4.6}$$

4.3 Tensor compressive sensing

4.3.1 Tensor compressive sensing with serial recovery

We first discuss our method for matrices, i.e. $d = 2$ and then for tensors $d \geq 3$.

Theorem 4.3.1 *Let $X = [x_{ij}] \in \mathbb{R}^{N_1 \times N_2}$ be s-sparse. Let $U_i \in \mathbb{R}^{m_i \times N_i}$ and assume that U_i has NSP$_s$ property for $i = 1, 2$. Define*

$$Y = [y_{pq}] = U_1 X U_2^\top \in \mathbb{R}^{m_1 \times m_2}. \tag{4.7}$$

Then X can be recovered uniquely using the following procedure. Let $\mathbf{y}_1, \ldots, \mathbf{y}_{m_2} \in \mathbb{R}^{m_1}$ be the columns of Y. Let $\mathbf{z}_i^\star \in \mathbb{R}^{N_1}$ be a solution of

$$\mathbf{z}_i^\star = \min\{\|\mathbf{z}_i\|_1,\ U_1 \mathbf{z}_i = \mathbf{y}_i\}, \quad i = 1, \ldots, m_2. \tag{4.8}$$

Then each $\mathbf{z}_i^\star$ is unique and s-sparse. Let $Z \in \mathbb{R}^{N_1 \times m_2}$ be the matrix with columns $\mathbf{z}_1^\star, \ldots, \mathbf{z}_{m_2}^\star$. Let $\mathbf{w}_1^\top, \ldots, \mathbf{w}_{N_1}^\top$ be the rows of Z. Then the j^{th} row of X is the solution $\mathbf{u}_j^\star \in \mathbb{R}^{N_2}$ of

$$\mathbf{u}_j^\star = \min\{\|\mathbf{u}_j\|_1,\ U_2 \mathbf{u}_j = \mathbf{w}_j\}, \quad j = 1, \ldots, N_1. \tag{4.9}$$

Proof Let $Z = X U_2^\top \in \mathbb{R}^{N_1 \times m_2}$. Assume that $\mathbf{z}_1^\star, \ldots, \mathbf{z}_{m_2}^\star$ are the columns of Z. Note that $\mathbf{z}_i^\star$ is a linear combination of the N_2 columns of X, given by the i^{th} row of U_2. Since X is s-sparse, $\mathbf{z}_i^\star$ has at most s nonzero entries. Note that $Y = U_1 Z$, it follows that $\mathbf{y}_i = U_1 \mathbf{z}_i^\star$. Since U_1 has NSP$_s$, we deduce the equality (Equation 4.8). Observe next that $Z^\top = U_2 X^\top$. Hence the column $\mathbf{w}_j$ of $Z^\top$ is $\mathbf{w}_j = U_2 \mathbf{u}_j^\star$. Since X is s-sparse, each $\mathbf{u}_j^\star$ is s-sparse. The assumption that U_2 has NSP$_s$ property implies (Equation 4.9). This completes the proof. ■

If we choose U_1, U_2 to be random, then we need the assumption (Equation 4.6). We now make the following observation. Suppose we know that each column of XU_2^T is s_1-sparse and each row of X is s_2-sparse. Then from the proof of Theorem 4.3.1 it follows that we can recover X, on the assumption that U_1 has NSP_{s_1} and U_2 has NSP_{s_2}.

Theorem 4.3.2 (GTCS-S) *Let $\mathcal{X} = [x_{i_1,\ldots,i_d}] \in \mathbb{R}^{N_1 \times \ldots \times N_d}$ be s-sparse. Let $U_i \in \mathbb{R}^{m_i \times N_i}$ and assume that U_i has NSP_s property for $i = 1, \ldots, d$. Define*

$$\mathcal{Y} = [y_{j_1,\ldots,j_d}] = \mathcal{X} \times_1 U_1 \times \ldots \times_d U_d \in \mathbb{R}^{m_1 \times \ldots \times m_d}. \tag{4.10}$$

Then $\mathcal{X}$ can be recovered uniquely using the following recursive procedure. Unfold $\mathcal{Y}$ in mode 1*,*

$$Y_{(1)} = U_1 X_{(1)} [\otimes_{k=d}^{2} U_k]^\top \in \mathbb{R}^{m_1 \times (m_2 \cdot \ldots \cdot m_d)}.$$

Let $\mathbf{y}_1, \ldots, \mathbf{y}_{m_2 \cdot \ldots \cdot m_d}$ be the columns of $Y_{(1)}$. Then $\mathbf{y}_i = U_1 \mathbf{z}_i$ where each $\mathbf{z}_i \in \mathbb{R}^{N_1}$ is s-sparse. Recover each $\mathbf{z}_i$ using (Equation 4.3)*. Let $\mathcal{Z} = \mathcal{X} \times_2 U_2 \times \ldots \times_d U_d \in \mathbb{R}^{N_1 \times m_2 \times \ldots \times m_d}$ with its mode-1 fibers being $\mathbf{z}_1, \ldots, \mathbf{z}_{m_2 \cdot \ldots \cdot m_d}$. Unfold $\mathcal{Z}$ in mode 2,*

$$Z_{(2)} = U_2 X_{(2)} [\otimes_{k=d}^{3} U_k \otimes I]^\top \in \mathbb{R}^{m_2 \times (N1 \cdot m_3 \cdot \ldots \cdot m_d)}.$$

Let $\mathbf{w}_1, \ldots, \mathbf{w}_{N1 \cdot m_3 \cdot \ldots \cdot m_d}$ be the columns of $Z_{(2)}$. Then $\mathbf{w}_j = U_2 \mathbf{v}_j$ where each $\mathbf{v}_j \in \mathbb{R}^{N_2}$ is s-sparse. Recover each $\mathbf{v}_j$ using (Equation 4.3)*. Continue the above procedure for mode $3, \ldots, d$ and $\mathcal{X}$ can be reconstructed in series.*

As for matrices, assume mode-i fibers of $\mathcal{X} \times_{i+1} U_{i+1} \times \ldots \times_d U_d$ is s_i-sparse for $i = 1, \ldots, d-1$ and mode-d fibers of $\mathcal{X}$ is s_d-sparse, then we can relax the condition such that U_i only has to satisfy NSP_{s_i} for $i = 1, \ldots, d$.

4.3.2 Tensor compressive sensing with parallel recovery

Theorem 4.3.3 (GTCS-P) *Let $\mathcal{X} = [x_{i_1,\ldots,i_d}] \in \mathbb{R}^{N_1 \times \ldots \times N_d}$ be s-sparse. Let $U_i \in \mathbb{R}^{m_i \times N_i}$ and assume that U_i has NSP_s property for $i = 1, \ldots, d$. Define $\mathcal{Y}$ as in* (Equation 4.10)*, then $\mathcal{X}$ can be recovered uniquely using the following procedure. Consider a decomposition of $\mathcal{Y}$ such that,*

$$\mathcal{Y} = \sum_{i=1}^{K} \mathbf{b}_i^{(1)} \circ \ldots \circ \mathbf{b}_i^{(d)}, \quad \mathbf{b}_i^{(j)} \in R_j(\mathcal{Y}) \subseteq U_j R_j(\mathcal{X}), j \in [d]. \tag{4.11}$$

Let $\mathbf{w}_i^{(j)\star} \in R_j(\mathcal{X}) \subset \mathbb{R}^{N_j}$ be a solution of

$$\mathbf{w}_i^{(j)\star} = \min\{\|\mathbf{w}_i^{(j)}\|_1,\ U_j \mathbf{w}_i^{(j)} = \mathbf{b}_i^{(j)}\}, \quad i = 1, \ldots, K, j = 1, \ldots, d. \tag{4.12}$$

Thus each $\mathbf{w}_i^{(j)\star}$ is unique and s-sparse. Then,

$$\mathcal{X} = \sum_{i=1}^{K} \mathbf{w}_i^{(1)} \circ \ldots \circ \mathbf{w}_i^{(d)}, \quad \mathbf{w}_i^{(j)} \in R_j(\mathcal{X}), j \in [d]. \tag{4.13}$$

Proof Since $\mathcal{X}$ is s-sparse, each vector in $R_j(\mathcal{X})$ is s-sparse. So if each U_j has NSP_s, we can get a unique s-sparse vector $\mathbf{w}_i^{(j)} \in R_j(\mathcal{X})$ such that $U_j \mathbf{w}_i^{(j)} = \mathbf{b}_i^{(j)}$ and obtain a tensor

$$\mathcal{Z} = \sum_{i=1}^{K} \mathbf{w}_i^{(1)} \circ \ldots \circ \mathbf{w}_i^{(d)}, \quad \mathbf{w}_i^{(j)} \in R_j(\mathcal{X}), j \in [d]. \tag{4.14}$$

We have $(\mathcal{X} - \mathcal{Z}) \times_1 U_1 \times \ldots \times_d U_d = 0$. We next show $\mathcal{Z} = \mathcal{X}$. For that we are going to assume slightly more general scenario. Namely each $R_j(\mathcal{X}) \subseteq \mathbf{V}_j \in \mathbb{R}^{N_j}$ such that each nonzero vector in $\mathbf{V}_j$ is s-sparse. So $R_j(\mathcal{Y}) \subseteq U_j R_j(\mathcal{X}) \subseteq U_j \mathbf{V}_j$ for $j \in [d]$. Assume $\mathcal{X} \neq \mathcal{Z}$. We next prove by induction on mode k which yields contradiction to this assumption.

When $k = 1$, unfold $\mathcal{X}$ and $\mathcal{Z}$ in mode 1 to obtain matrices $X_{(1)}$ and $Z_{(1)}$. The column space of $X_{(1)}$ and $Z_{(1)}$ are contained in $\mathbf{V}_1$ while the row spaces are contained in $\hat{\mathbf{V}}_1 := \mathbf{V}_2 \circ \ldots \circ \mathbf{V}_d$. Since we assume that $\mathcal{X} \neq \mathcal{Z}$, thus $X_{(1)} - Z_{(1)} \neq 0$. Then $X_{(1)} - Z_{(1)} = \sum_{i=1}^{p} \mathbf{u}_i \mathbf{v}_i^\top$ where rank $(X_{(1)} - Z_{(1)}) = p$, $\mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathbf{V}_1, \mathbf{v}_1, \ldots, \mathbf{v}_p \in \hat{\mathbf{V}}_1$ are linearly independent. Observe next that $U_1\mathbf{u}_1, \ldots, U_1\mathbf{u}_p$ are linearly independent. To show this, $\mathbf{0} = \sum_{i=1}^{p} a_i U_1 \mathbf{u}_i = U_1 \mathbf{u}, \quad \mathbf{u} = \sum_{i=1}^{p} a_i \mathbf{u}_i \in \mathbf{V}_1$. Since $\mathbf{u}$ is s-sparse and U_1 has NSP_s we deduce that $\mathbf{u} = \mathbf{0}$, so $a_1 = \ldots = a_p = 0$.

Since $(\mathcal{X} - \mathcal{Z}) \times_1 U_1 \times \ldots \times_d U_d = 0$, it is equivalent to

$$0 = U_1(X_{(1)} - Z_{(1)})(U_d \otimes \ldots \otimes U_2)^\top = U_1(X_{(1)} - Z_{(1)})\hat{U}_1^\top = \sum_{i=1}^{p} (U_1 \mathbf{u}_i)(\hat{U}_1 \mathbf{v}_i)^\top.$$

Since $U_1\mathbf{u}_1, \ldots, U_1\mathbf{u}_p$ are linearly independent it follows that $\hat{U}_1\mathbf{v}_i = \mathbf{0}$ for $i = 1, \ldots, p$. Therefore,

$$(X_{(1)} - Z_{(1)})\hat{U}_1^\top = (\sum_{i=1}^{p} \mathbf{u}_i\mathbf{v}_i^\top)\hat{U}_1^\top = \sum_{i=1}^{p} \mathbf{u}_i(\hat{U}_1\mathbf{v}_i)^\top = 0.$$

Fold back to tensor form, this is equivalent to

$$(\mathcal{X} - \mathcal{Z}) \times_1 I \times_2 U_2 \times \ldots \times_d U_d = (\mathcal{X} - \mathcal{Z}) \times_2 U_2 \times \ldots \times_d U_d = 0. \tag{4.15}$$

Suppose when $k = m$, we have $(\mathcal{X} - \mathcal{Z}) \times_m U_m \times \ldots \times_d U_d = 0$. Continue to unfold $\mathcal{X}$ and $\mathcal{Z}$ in mode m, The column space of $X_{(m)}$ and $Z_{(m)}$ are contained in $\mathbf{V}_m$ while the row spaces are contained in $\hat{\mathbf{V}}_m := \mathbf{V}_1 \circ \ldots \circ \mathbf{V}_{m-1} \circ \mathbf{V}_{m+1} \circ \ldots \circ \mathbf{V}_d$. Since we assume that $\mathcal{X} \neq \mathcal{Z}$, thus $X_{(m)} - Z_{(m)} \neq 0$. Then $X_{(m)} - Z_{(m)} = \sum_{i=1}^{q} \mathbf{f}_i\mathbf{g}_i^\top$ where rank $(X_{(m)} - Z_{(m)}) = q$, $\mathbf{f}_1, \ldots, \mathbf{f}_q \in \mathbf{V}_m, \mathbf{g}_1, \ldots, \mathbf{g}_q \in \hat{\mathbf{V}}_m$ are linearly independent. Observe next that $U_m\mathbf{f}_1, \ldots, U_m\mathbf{f}_q$ are linearly independent.

Since $(\mathcal{X} - \mathcal{Z}) \times_m U_m \times \ldots \times_d U_d = 0$, it is equivalent to

$$0 = U_m(X_{(m)} - Z_{(m)})(U_d \otimes \ldots \otimes U_{m+1} \otimes I)^\top = U_m(X_{(m)} - Z_{(m)})\hat{U}_m^\top = \sum_{i=1}^{q} (U_m\mathbf{f}_i)(\hat{U}_m\mathbf{g}_i)^\top.$$

Since $U_m\mathbf{f}_1, \ldots, U_m\mathbf{f}_q$ are linearly independent it follows that $\hat{U}_m\mathbf{g}_i = \mathbf{0}$ for $i = 1, \ldots, q$. Therefore,

$$(X_{(m)} - Z_{(m)})\hat{U}_m^\top = (\sum_{i=1}^{q} \mathbf{f}_i\mathbf{g}_i^\top)\hat{U}_m^\top = \sum_{i=1}^{q} \mathbf{f}_i(\hat{U}_m\mathbf{g}_i)^\top = 0.$$

Fold back to tensor form, this is equivalent to

$$(\mathcal{X} - \mathcal{Z}) \times_1 I \times \ldots \times_m I \times_{m+1} U_{m+1} \times \ldots \times_d U_d = (\mathcal{X} - \mathcal{Z}) \times_{m+1} U_{m+1} \times \ldots \times U_d = 0.$$

Consequently, when $m = d$, by unfolding $\mathcal{X}$ and $\mathcal{Z}$ in mode d, we will derive that

$$\mathcal{X} - \mathcal{Z} = 0.$$

This brings contradiction to our assumption that $\mathcal{X} \neq \mathcal{Z}$. Thus, it proves that $\mathcal{X} = \mathcal{Z}$. This completes the proof. ■

In fact, if all vectors $\in R_i(\mathcal{X})$ are s_i-sparse, then U_i only has to satisfy NSP_{s_i}. The above recovery procedure consists of two stages: the decomposition stage and the reconstruction stage where the latter can be implemented in parallel.

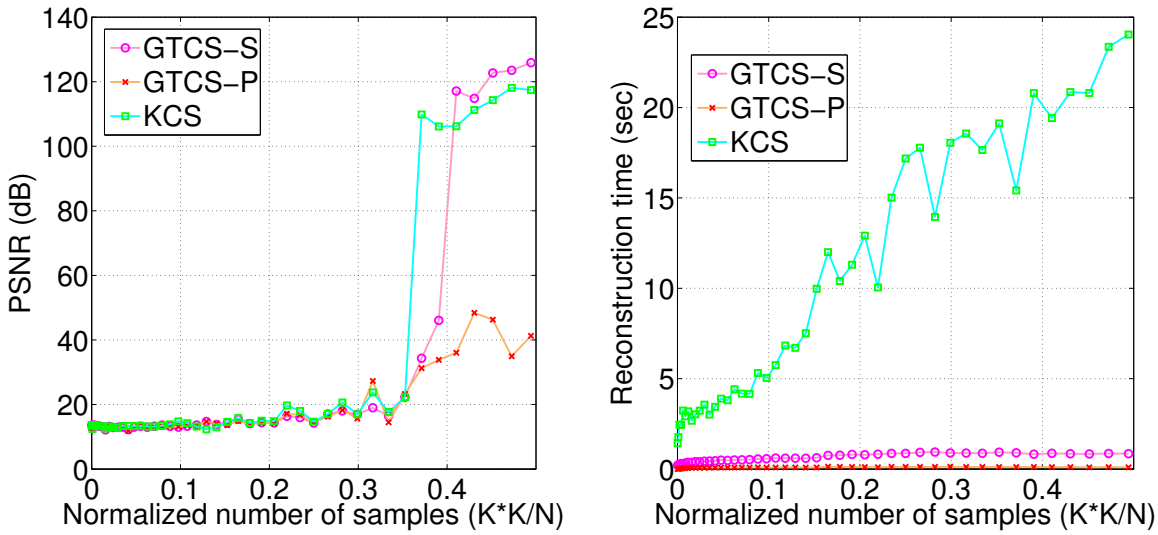
4.4 Experimental Results

We experimentally demonstrate the performance of GTCS methods on sparse image and video sequence. In (46), KCS has shown its outstanding performance for compression of multidimensional signals in comparison with several other methods such as independent measurements and partitioned measurements. Therefore, we choose KCS as a comparison to the proposed GTCS methods. Our experiments use the ℓ_1-minimization solvers from (48). We set the same threshold to determine the termination of ℓ_1-minimization in all subsequent experiments. All simulations are executed on a desktop with 2.4 GHz Intel Core i5 CPU and 8GB RAM.

4.4.1 Sparse image representation

As shown in Figure 11(a), the original black and white image is of size 64×64 ($N = 4096$ pixels). Its columns are 14-sparse and rows are 18-sparse. The image itself is 178-sparse. We let the number of measurements evenly split among the two modes, that is, for each mode, the randomly constructed Gaussian matrix U is of size $K \times 64$. Therefore the KCS measurement matrix $U \otimes U$ is of size $K^2 \times 4096$. Thus the total number of samples is K^2. We define the normalized number of samples to be $\frac{K^2}{N}$. For GTCS, both the serial recovery method GTCS-S and the parallelizable recovery method GTCS-P are implemented. In the matrix case, we simply conduct SVD on the compressed image in the decomposition stage of GTCS-P. Although the reconstruction stage of GTCS-P is parallelizable, we here recover each vector in series. We comprehensively examine the performance of all the above methods by varying K from 1 to 45.

Figure 10(a) and 10(b) compare the peak signal to noise ratio (PSNR) and the recovery time respectively. Both KCS and GTCS methods achieve PSNR over 30dB when $K = 39$. As K increases, GTCS-S tends to outperform KCS in terms of both accuracy and efficiency. Although PSNR of GTCS-P is the lowest among the three methods, it is most time efficient. Moreover, with parallelization of GTCS-P, the recovery procedure can be further accelerated considerably. The reconstructed images when $K = 38$, that is, using 0.35 normalized number of samples, are shown in Figure 11(b)11(c)11(d). Though GTCS-P recovers much noisier image, it is good at recovering the non-zero structure of the original image.



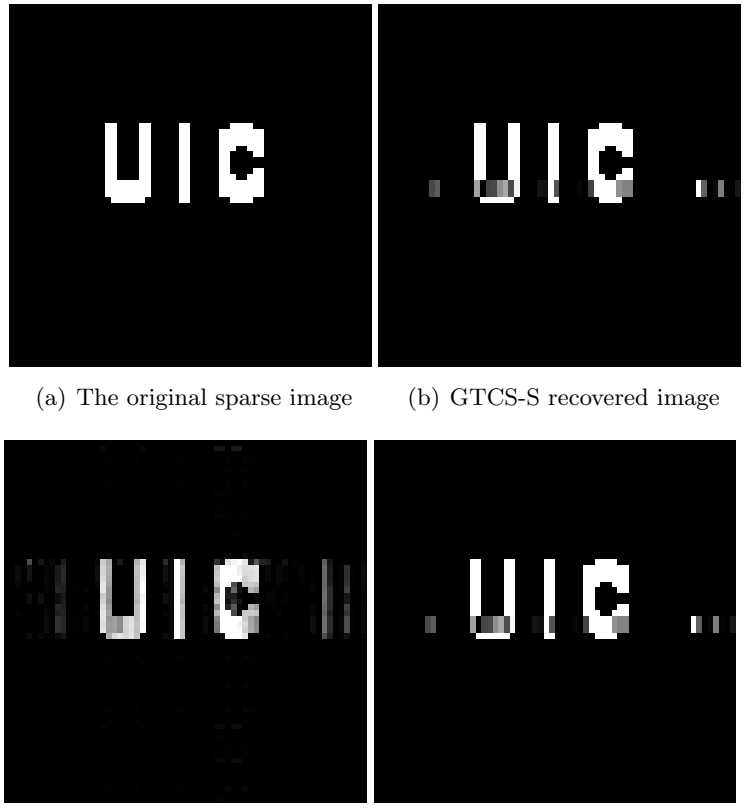
(a) PSNR comparison

(b) Recovery time comparison

Figure 10. PSNR and reconstruction time comparison on sparse image.

4.4.2 Sparse video representation

We next compare the performance of GTCS and KCS on video data. Each frame of the video sequence is preprocessed to have size 24×24 and we choose the first 24 frames. The video data together is represented by a $24 \times 24 \times 24$ tensor and has $N = 13824$ voxels in total. To obtain a sparse tensor, we manually keep only $6 \times 6 \times 6$ nonzero entries in the center of the video tensor data and the rest are set to zero. Therefore, the video tensor itself is 216-sparse and its mode-i fibers are all 6-sparse for $i = 1, \ldots, 3$. The randomly constructed Gaussian measurement matrix for each mode is now of size $K \times 24$ and the total number of samples is K^3. Therefore, the normalized number of samples is $\frac{K^3}{N}$. In the decomposition stage of



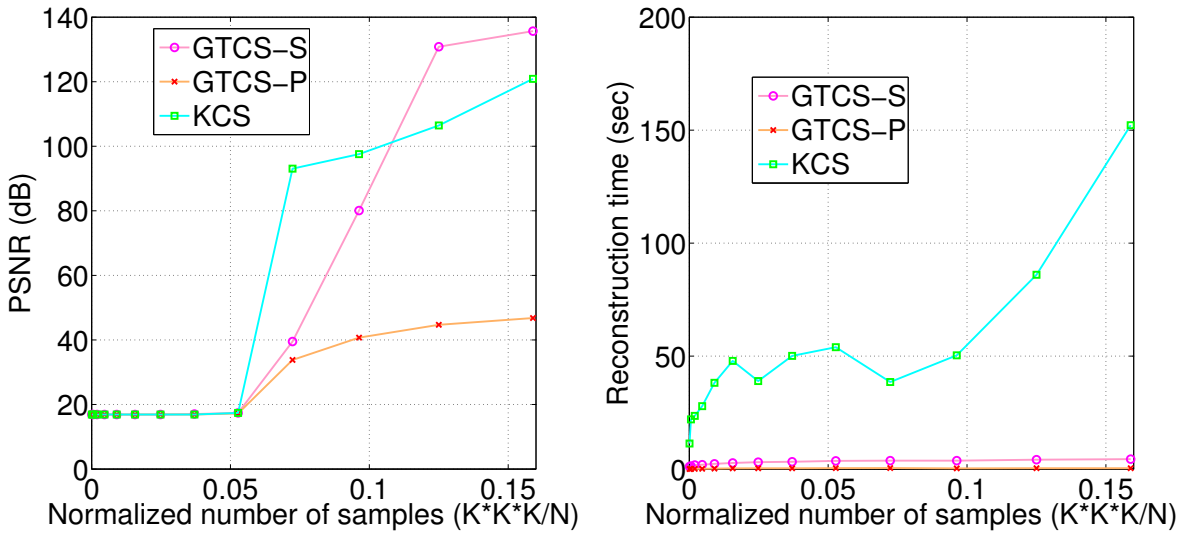
(a) The original sparse image (b) GTCS-S recovered image

(c) GTCS-P recovered image (d) KCS recovered image

Figure 11. The original image and the recovered images by GTCS and KCS using 0.35 normalized number of samples.

GTCS-P, we employ a decomposition described in Section 4.2.1 to obtain a weaker form of the core Tucker decomposition. We vary K from 1 to 13.

Figure 12(a) depicts PSNR of the first non-zero frame recovered by all three methods. All methods exhibit an abrupt increase in PSNR at $K = 10$ (using 0.07 normalized number of samples). Also, Figure 12(b) summarizes the recovery time. In comparison to the image case,



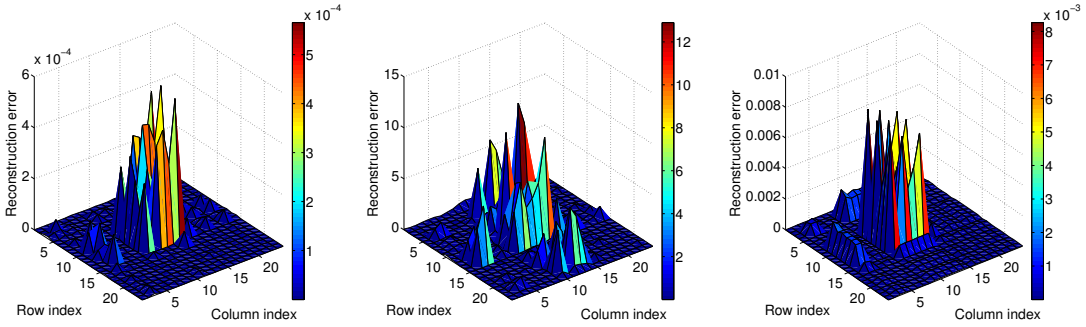
(a) PSNR comparison

(b) Recovery time comparison

Figure 12. PSNR and reconstruction time comparison on sparse video.

the time advantage of GTCS becomes more important in the reconstruction of higher-order tensor data.

We specifically look into the recovered frames of all three methods when $K = 12$. Since all the recovered frames achieve a PSNR higher than 40 dB, it is hard to visually observe any difference compared to the original video frame. Therefore, we display the reconstruction error image defined as the absolute difference between the reconstructed image and the original image. Figures 13(a)13(b)13(c) visualize the reconstruction errors of all three methods. Compared to KCS, GTCS-S achieves much lower reconstruction error using much less time.



(a) Reconstruction error of GTCS-S (b) Reconstruction error of GTCS-P (c) Reconstruction error of KCS

Figure 13. Visualization of the reconstruction error in the recovered video frame 9 using 0.125 normalized number of samples.

4.5 Conclusion

In this paper, we propose Generalized Tensor Compressive Sensing (GTCS)–a unified framework for compressive sensing of higher-order tensors. In addition, we propose two reconstruction procedures, a serial method (GTCS-S) and a parallelizable method (GTCS-P). We then compare the performance of the proposed method with Kronecker compressive sensing (KCS) on both image and video data. Experimental results show that GTCS outperforms KCS in terms of both accuracy and efficiency. The advantage of our method mainly comes from the fact that our recovery problems are in terms of each tensor mode, which are much smaller comparing with the recovery related to the vectorization of all tensor modes. Such advantage becomes more important as the order of the data increases. We state our theorems for sparse tensors. However, most real-world data are not really sparse, instead, they are only compressible in some

domain. Future work will focus on demonstrating the effectiveness of GTCS on compressible higher-order data.

CHAPTER 5

CONCLUSION

This thesis summarized our research in the application of multilinear algebra to higher-order data analysis including retrieval, classification and representation. It mainly consisted of a HOSVD-based multilinear indexing and retrieval approach of multifactor data in Chapter 2, a multilinear extension of LDA for higher-order data classification in Chapter 3 and a unified framework for compressive sensing of higher-order tensors which integrates acquisition and compression from all tensor modes in Chapter 4. We discussed these algorithms in detail and demonstrated their superior performance through exhaustive computer simulations, compared to existing methods.

Finally, there are still some aspects of the proposed methods that deserve further study. In all the proposed methods, we assumed that the correspondence of the tensor modes in all data is known whereas in most real applications, the correspondence between the tensor structures is unknown. Future work will examine the performance of the proposed methods without such correspondence. In Chapter 3, we only proved that the objective function sequence generated by CMDA iterative procedure is asymptotically bounded. We are not sure if such procedure always converges. If not, what factors affect its convergency? Besides, both DGTDA and CMDA are multilinear, they can not capture the nonlinear structure of the data manifold effectively. It remains unclear how to generalize our approach to nonlinear case. In Chapter 4, we stated our theorems for sparse tensors. However, most real-world data are not really sparse, instead,

they are only compressible in some domain. Future work will also look into the effectiveness of GTCS on compressible higher-order data.

CITED LITERATURE

1. Shashua, A. and Levin, A.: Linear image coding for regression and classification using the tensor-rank principle. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–42–I–49 vol.1, 2001.

2. Vasilescu, M. A. O. and Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In IN PROCEEDINGS OF THE EUROPEAN CONFERENCE ON COMPUTER VISION, pages 447–460, 2002.

3. Ma, X., Schonfeld, D., and Khokhar, A.: Dynamic updating and downdating matrix svd and tensor hosvd for adaptive indexing and retrieval of motion trajectories. In Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pages 1129–1132, 2009.

4. Tao, D.: Discriminative linear and multilinear subspace methods. Doctoral dissertation, University of London, 2006.

5. The context aware vision using image-based active recognition (caviar) dataset. In Available: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/, URL.

6. Duda, R. O., Hart, P. E., and Stork, D. G.: Pattern Classification. New York, N.Y., Wiley-Interscience, second edition, 2000.

7. Turk, M. and Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71–86, jan 1991.

8. Belhumeur, P., Hespanha, J., and Kriegman, D.: Eigenfaces vs. fisherfaces: recognition using class specific linear projection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 19(7):711–720, jul 1997.

9. Swets, D. and Weng, J.: Using discriminant eigenfeatures for image retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 18(8):831–836, aug 1996.

CITED LITERATURE (Continued)

10. Shashua, A. and Levin, A.: Linear image coding for regression and classification using the tensor-rank principle. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–42–I–49, 2001.

11. Yang, J., Zhang, D., Frangi, A., and Yang, J.: Two-dimensional pca: a new approach to appearance-based face representation and recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 26(1):131–137, jan 2004.

12. Liu, K., Cheng, Y., and Yang, J.: Algebraic feature extraction for image recognition based on an optimal discriminant criterion. Pattern Recognition, 26(6):903–911, 1993.

13. Kong, H., Teoh, E., Wang, J.-G., and Venkateswarlu, R.: Two dimensional fisher discriminant analysis: Forget about small sample size problem. In Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, volume 2, pages 761–764, 18-23.

14. Ye, J., Janardan, R., and Li, Q.: Two-dimensional linear discriminant analysis. In Advances in Neural Information Processing Systems 17, pages 1569–1576. Cambridge, MA, MIT Press, 2005.

15. He, X., Cai, D., and Niyogi, P.: Tensor subspace analysis. In Advances in Neural Information Processing Systems 18, pages 499–506. Cambridge, MA, MIT Press, 2006.

16. Cai, D., He, X., and Han, J.: Subspace learning based on tensor analysis. Computer Science Research and Tech Reports, may 2005.

17. Vasilescu, M. and Terzopoulos, D.: Multilinear subspace analysis of image ensembles. In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, volume 2, pages II – 93–9 vol.2, jun 2003.

18. Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X., and Zhang, H.-J.: Discriminant analysis with tensor representation. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 526–532, jun 2005.

19. Tao, D., Li, X., Wu, X., and Maybank, S.: General tensor discriminant analysis and gabor features for gait recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(10):1700–1715, oct 2007.

CITED LITERATURE (Continued)

20. Li, Q., Shi, X., and Schonfeld, D.: A general framework for robust hosvd-based indexing and retrieval with high-order tensor data. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pages 873–876, may 2011.

21. Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N.: A survey of multilinear subspace learning for tensor data. Pattern Recognition, 44(7):1540–1551, July 2011.

22. Wang, Y. and Gong, S.: Tensor discriminant analysis for view-based object recognition. In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, volume 3, pages 33–36, 2006.

23. Tao, D., Li, X., Wu, X., and Maybank, S.: Elapsed time in human gait recognition: A new approach. In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, volume 2, pages 177–180, May.

24. Shashua, A. and Levin, A.: Linear image coding for regression and classification using the tensor-rank principle. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–42–I–49 vol.1.

25. Lu, H., Plataniotis, K., and Venetsanopoulos, A.: Uncorrelated multilinear discriminant analysis with regularization and aggregation for tensor object recognition. Neural Networks, IEEE Transactions on, 20(1):103–123, Jan.

26. Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X., and Zhang, H.-J.: Multilinear discriminant analysis for face recognition. Image Processing, IEEE Transactions on, 16(1):212–220, Jan.

27. Kolda, T. G. and Bader, B. W.: Tensor decompositions and applications. SIAM REVIEW, 51(3):455–500, 2009.

28. Fukunaga, K.: Introduction to Statistical Pattern Classification. San Diego, CA., Academic Press, second edition, 1990.

29. Georghiades, A., Belhumeur, P., and Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intelligence, 23(6):643–660, 2001.

CITED LITERATURE (Continued)

30. Candes, E., Romberg, J., and Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. Information Theory, IEEE Transactions on, feb. 2006.

31. Candes, E. and Tao, T.: Near-optimal signal recovery from random projections: Universal encoding strategies? Information Theory, IEEE Transactions on, 52:5406–5425, dec. 2006.

32. Donoho, D.: Compressed sensing. Information Theory, IEEE Transactions on, apr. 2006.

33. Donoho, D. and Logan, B.: Signal recovery and the large sieve. SIAM Journal on Applied Mathematics, 52:577–591, 1992.

34. Chen, S., Donoho, D., and Saunders, M.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20:33–61, 1996.

35. Rudin, L., Osher, S., and Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60:259–268, 1992.

36. Cohen, A., Dahmen, W., and Devore, R.: Compressed sensing and best k-term approximation. J. Amer. Math. Soc, pages 211–231, 2009.

37. Candes, E.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique, 346:589–592, 2008.

38. Wakin, M., Laska, J., Duarte, M., Baron, D., Sarvotham, S., Takhar, D., Kelly, K., and Baraniuk, R.: An architecture for compressive imaging. In IEEE ICIP, pages 1273–1276, oct. 2006.

39. Candes, E. J., Romberg, J. K., and Tao, T.: . Comm. Pure Appl. Math., aug.

40. Gan, L.: Block compressed sensing of natural images. In 15th ICDSP 2007, pages 403–406, jul. 2007.

41. Fazel, M.: Matrix rank minimization with applications. Doctoral dissertation, 2002.

42. Recht, B., Fazel, M., and Parrilo, P.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev., 52.

CITED LITERATURE (Continued)

43. Candes, E. and Recht, B.: Exact matrix completion via convex optimization. Commun. ACM, jun. 2012.

44. Lim, L. and Comon, P.: Multiarray signal processing: Tensor decomposition meets compressed sensing. CoRR, abs/1002.4935, 2010.

45. Duarte, M. and Baraniuk, R.: Kronecker product matrices for compressive sensing. In IEEE ICASSP 2010, pages 3650–3653, mar. 2010.

46. Duarte, M. and Baraniuk, R.: Kronecker compressive sensing. Image Processing, IEEE Transactions on, 21:494–504, feb. 2012.

47. Kolda, T. G. and Bader, B. W.: Tensor decompositions and applications. SIAM REVIEW, 51, 2009.

48. Candes, E. J. and Romberg, J. K.: The l_1 magic toolbox. Available online: http://www.l1-magic.org.

49. Shannon, C.: Communication in the presence of noise. Proceedings of the IRE, 37:10–21, jan. 1949.

50. Vaughan, R., Scott, N., and White, D.: The theory of bandpass sampling. Signal Processing, IEEE Transactions on, 39:1973–1984, sep. 1991.

51. Oseledets, I.: A new tensor decomposition. Doklady Mathematics, 80:495–496, 2009.

52. Bader, B. W., Kolda, T. G., et al.: Matlab tensor toolbox version 2.5. Available online: http://www.sandia.gov/ tgkolda/TensorToolbox/, jan. 2012.

53. Kolda, T. G. and Sun, J.: Scalable tensor decompositions for multi-aspect data mining. In 8th IEEE ICDM 2008 Proceedings, pages 363–372, dec. 2008.

54. Li, Q.: Multilinear Algebra in High-Order Data Analysis: Retrieval, Classification and Representation. Doctoral dissertation, 2013.

55. Sidiropoulos, N. and Kyrillidis, A.: Multi-way compressed sensing for sparse low-rank tensors. Signal Processing Letters, IEEE, 19, 2012.

VITA

NAME	Qun Li
EDUCATION	Ph.D., Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, Illinois, 2013
	M.S., Electrical and Computer Engineering, University of Illinois at Chicago (UIC), Chicago, Illinois, 2012
	B.S., Communications Engineering, Nanjing University of Science and Technology, Nanjing, China, 2009
EXPERIENCE	Research Scientist, Xerox Research Center Webster, Webster, NY, 05/2012-07/2012
	Research Assistant, Multimedia Communications Laboratory, Dept. of ECE, University of Illinois at Chicago, 08/2010-12/2012
	Teaching Assistant, Dept. of ECE, University of Illinois at Chicago, 08/2009-05/2013
	Teaching Assistant, Department of Electrical and Computer Engineering, UIC, Chicago, IL, USA, 08/2007 - 05/2008
PUBLICATIONS	Q. Li, D. Schonfeld: Multilinear discriminant analysis for higher-order tensor data classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, submitted.
	Q. Li, X. Shi and D. Schonfeld: HOSVD-based higher-order data indexing and retrieval. IEEE Signal Processing Letters, submitted.
	Q. Li, D. Schonfeld and S. Friedland: Generalized tensor compressive sensing. IEEE International Conference on Multimedia and Expo (ICME), San Jose, CA, 2013.

VITA (Continued)

Q. Li, X. Shi and D. Schonfeld: A general framework for robust HOSVD-based indexing and retrieval with high-order tensor data. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Prague, Czech Republic, 2011.

Q. Li, X. Shi and D. Schonfeld: Robust HOSVD-based multi-camera motion trajectory indexing and retrieval. SPIE Electronic Imaging (EI), San Francisco, CA, 2011.

PATENT — Q. Li, E. Bernal, W. Wu, B. Xu, R. Loce, R. Bala and S. Schweid: Methods and systems for multimedia trajectory annotation.

INVITED TALK — S. Friedland, D. Schonfeld and Q. Li: Generalized tensor compressive sensing, Matheon Workshop Compressed Sensing and its Applications, Dec 2013, Berlin, Germany.