Advanced Multilinear Data Analysis and Sparse Representation Approaches and Their Applications

Khoa Luu

A Thesis
In The Department of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy in Computer Science
Concordia University
Montreal, Quebec, Canada

November 2013

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Khoa Luu

Entitled: Advanced Multilinear Data Analysis and Sparse Representation Approaches and Their Applications

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Prof. Sudhir Mudur — Chair

Prof. Sri Krishnan (Ryerson University) — External Examiner

_____ — External to Program

Prof. Adam Krzyzak — Examiner

Prof. Wei-Ping Zhu — Examiner

Profs. Tien D. Bui, Ching Y. Suen and Marios Savvides — Thesis Supervisor

Approved by

_____
Chair of Department or Graduate Program Director

Nov. 3, 3013

_____
Dean of Faculty

Advanced Multilinear Data Analysis and Sparse Representation Approaches and Their Applications

Khoa Luu

Concordia University, 2013

Abstract

Multifactor analysis plays an important role in data analysis since most real-world datasets usually exist with a combination of numerous factors. These factors are usually not independent but interdependent together. Thus, it is a mistake if a method only considers one aspect of the input data while ignoring the others. Although widely used, Multilinear PCA (MPCA), one of the leading multilinear analysis methods, still suffers from three major drawbacks. Firstly, it is very sensitive to outliers and noise and unable to cope with missing values. Secondly, since MPCA deals with huge multidimensional datasets, it is usually computationally expensive. Finally, it loses original local geometry structures due to the averaging process. This thesis sheds new light on the tensor decomposition problem via the ideas of fast low-rank approximation in random projection and tensor completion in compressed sensing. We propose a novel approach called Compressed Submanifold Multifactor Analysis (CSMA) to solve the three problems mentioned above. Our approach is able to deal with the problem of missing values and outliers via our proposed novel sparse Higher-order Singular Value Decomposition approach, named HOSVD-L1 decomposition. The Random Projection method is used to obtain the fast low-rank approximation of a given multifactor dataset. In addition, our method can preserve geometry of the original data.

In the second part of this thesis, we present a novel pattern classification approach named Sparse Class-dependent Feature Analysis (SCFA), to connect the advantages of sparse representation in an overcomplete dictionary, with a powerful nonlinear classifier. The classifier is based on the estimation of class-specific optimal filters, by solving an L1-norm optimization problem using the Alternating Direction Method of Multipliers.

Our method as well as its Reproducing Kernel Hilbert Space (RKHS) version is tolerant to the presence of noise and other variations in an image. Our proposed methods achieve very high classification accuracies in face recognition on two challenging face databases, i.e. the CMU Pose, Illumination and Expression (PIE) database and the Extended YALE-B that exhibit pose and illumination variations; and the AR database that has occluded images. In addition, they also exhibit robustness on other evaluation modalities, such as object classification on the Caltech101 database. Our method outperforms state-of-the-art methods on all these databases and hence they show their applicability to general computer vision and pattern recognition problems.

Thesis Supervisors: Tien D. Bui (Concordia University)
Title: Professor

Thesis Supervisors: Marios Savvides (Carnegie Mellon University)
Title: Professor

Thesis Supervisors: Ching Y. Suen (Concordia University)
Title: Professor

Dedication

To my parents, parents-in-law, and my wife who provided me with the motivation to complete my Ph.D. study. If it had not been for their love and support, I could not have completed this thesis.

Acknowledgements

None of my thesis work could have been completed without the support and dedication of numerous people. First of all, I would especially like to thank my supervisors, Prof. Tien Dai Bui and Prof. Ching Y. Suen, who have given me an unforgettable impression with their erudition and very kind manner. Although having tons of work, they always tried to spend their "epsilon" available time on encompassing my studies and listening to all of my ideas. I also would like to give my thanks to Prof. Marios Savvides for providing me with a unique opportunity and an ideal environment to studies the topics fascinated me profoundly. During my Ph.D. studies and my work, no matter what research topics, he has always supported me and encouraged my freedom in research. From him, I have learned the values of hard work and enthusiasm for both research work and practical experience in industrial projects.

I am also indebted to my colleagues at CMU Cylab Biometrics Center and CENPARMI who have been very kind and friendly. At the CMU Cylab Biometrics Center, I have been lucky to work in a professional research environment. I have had the chance to attend to talks by world-class researchers and professors and got some breakthrough ideas from fruitful discussions with them. I would like to thank my colleagues, Dr. Sung Won Park, Dr. Jingu Heo, Dr. Ramzi Abiantun, Keshav Seshadri, Shreyas Venugopalan, Felix Juefei Xu and Utsav Prabhu who have always been eager to help me with their best support not only in work but also in life since I arrived in Pittsburgh. Additionally, I would like to thank Keshav for his wonderful editorial assistance with this thesis. At CENPARMI, I would like to acknowledge Mr. Nicola Nobile, who is always keen to help all the lab members.

I would like to thank Profs. Sudhir Mudur, Adam Krzyzak, Wei-Ping Zhu and Sri Krishnan for reading my thesis during their busiest time. In addition, I would also like to give my thanks to many researchers all over the world, i.e., Dr. Sung Won Park at CMU Cylab Biometrics Center, Prof. Aswin Sankaranarayanan at Carnegie Mellon University, Prof. Haiping Lu at Hong Kong Baptist University (previously in University of Toronto), Yinqiang Zheng at the Tokyo Institute

of Technology, who were patient when explaining their work and who always gave me useful suggestions.

Finally, my deepest gratitude goes to my grandmothers, parents and especially my wife, Ngan Le, who is also my greatest colleague. These supporters always give me their total encouragement, endless support, and fruitful advice to keep me healthy, both at home and mostly abroad.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the well-known list of the "Top 10 algorithms" that have had the greatest influence on the development and practice of science and engineering during the 20th century [1, 28], we can find three entries that have a very close relationship to *matrix decomposition* problem, including the QR algorithm [82], the decompositional approach to matrix computation [98] and Krylov subspace iteration [27]. As Stewart explained in [98], the principal purpose of matrix decomposition and computation methods is not only to solve any particular problem but also to construct generally computational platforms that have the ability to solve the variations of these problems flexibly. When it happens, these methods can be then translated and emerged easily in computer hardware and distributed widely in numerous fields with more applications.

The problem of data decomposition becomes more interesting and challenging when the dimensionality of input data is increased. Then, instead of dealing with matrices, we now have to face up to *tensors* and *multiple factor* data. The purpose of this thesis is to present a novel tensor decomposition method that has the ability to analyze any given multifactor data. In addition, we also introduce a new pattern classification method that allows us to achieve very high classification accuracy on challenging databases with different modalities. The rest of this chapter is organized as follows. In the first section of this chapter,

we show the motivation for our discussed problem. Then, in the second section, we review some other standard deterministic tensor decomposition algorithms and discuss their limitations. The third section then shows our contributions in this thesis. Finally, we summary the thesis organization and the notations used within this thesis in the last sections.

1.1 Motivation

Dimensionality reduction is the process of reducing the number of variables or dimensions from a given high-dimensional data into a low-dimensional one. Since a digital image is a numeric representation of high-dimensional pixel values, it therefore can be analyzed efficiently via dimensionality reduction approaches. In addition, these methods can also remove unnecessary components and only keep the critical features in the analyzed data. There are numerous dimensionality reduction methods listed in this area, to name a few, Principal Component Analysis (PCA) [55, 105], Linear Discriminant Analysis (LDA) [8], Unsupervised Discriminant Projection (UDP) [121], Isomap [102], Locally Linear Embedding (LLE) [89, 91], Laplacian Eigenmaps [9], Neighborhood Preserving Embedding (NPE) [50], Linearity Preserving Projection (LPP) [51], etc. Figure 1.1 shows some examples of the dimensionality reduction methods mentioned above.

In the real world, however, data analyzed usually exists under the combination of numerous factors, particularly for facial images. Those facial images vary significantly due to numerous factors such as pose, illumination or subject identity [10, 70]. In addition, if soft-biometrics information is being studied, the use of facial images can also provide age, gender and ethnicity information of the given subjects. Therefore, in facial image analysis, dimensionality reduction approaches not only aim to reduce the number of dimensions of given high-dimensional data, but to also study the relationships among the different factors. For example, in the soft-biometrics problem [118], i.e., determination of age, gender and ethnicity of a subject in a given image, the method has to have the ability to extract the age

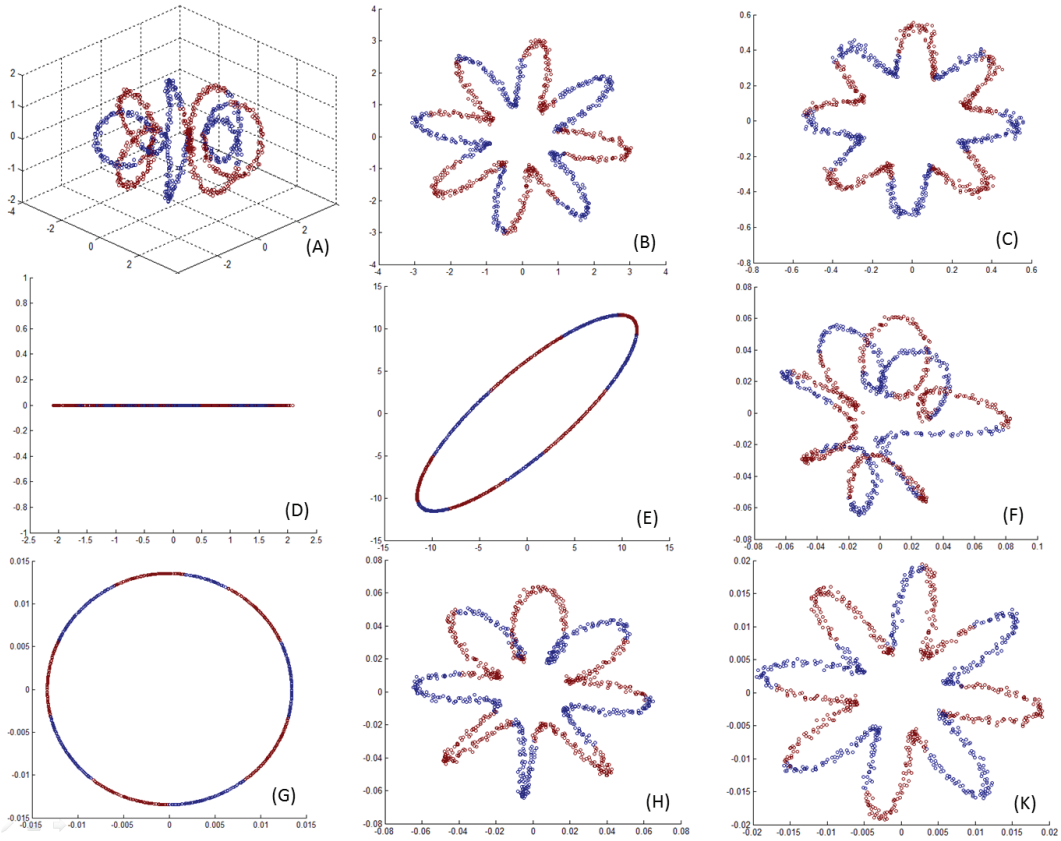


Figure 1.1: Dimensionality reduction with various approaches [106, 107]: (A) Original data distribution in 3 dimensions, (B) Principal Component Analysis, (C) Kernel PCA, (D) Linear Discriminant Analysis, (E) Isomap, (F) Locally Linear Embedding, (G) Laplacian Eigenmaps, (H) Neighborhood Preserving Embedding and (K) Linearity Preserving Projection.

information regardless of gender and ethnicity factors. Figure 1.2 shows the drawbacks of a commercial face recognition system that doesn't consider the relationships among the factors [2]. The face matching scores of two given subjects are dramatically dropped under different lighting conditions in the probe images.

Although the topic of multiple factor analysis and tensor decompositions has been studied actively for the past four decades in applied mathematics area, i.e. decompositions in data arrays [58, 104], it is still a rather new topic in image analysis and computer vision [40, 43, 47, 63, 67, 81, 108]. However, these methods are still very limited in representing data structures and are also unable to handle multi-dimensional data with missing values. With the recent fast development of compressed sensing techniques, there is a need to

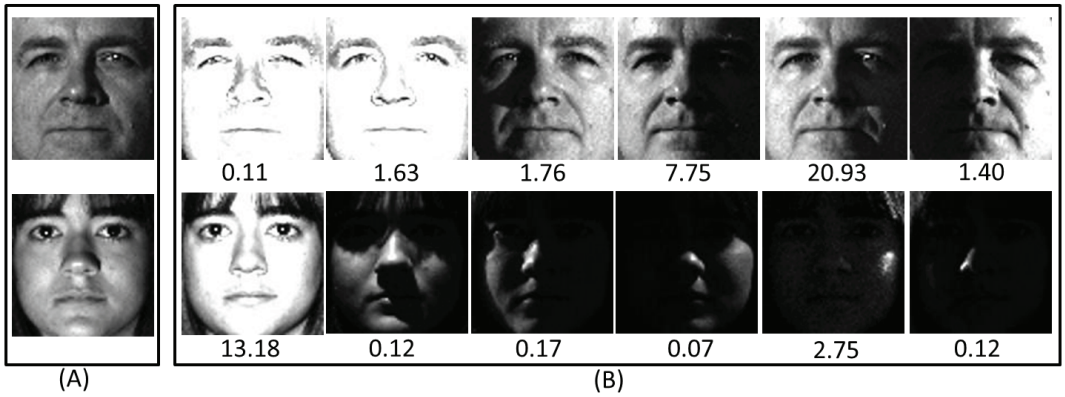


Figure 1.2: An example to show how face matching scores fall dramatically due to lighting variations when a commercial Face Recognition system [2] is used. (A): Gallery facial images, (B): Probe facial images. The numbers are the matching scores produced by the commercial Face Recognition system [2].

develop a new multiple factor analysis approach that can benefit from the efficiency of compressed sensing. Therefore, this thesis proposes a novel method for dimensionality reduction based on multiple factor analysis and applies this method to the problem of face recognition.

1.2 Multiple Factor Analysis

As discussed in Section 1.1, multifactor analysis plays an important role in data analysis since most real-world datasets usually exist with a combination of various factors. These factors are not independent but *interdependent*. In other words, relationships always exist among analyzing factors in real-world datasets. Therefore, it is not a good idea if a face recognition system focuses on only the subject identification factor and disregards all the other factors [127]. For example, given facial images as shown in Figure 1.3, several factors can be extracted, such as subject identity, illumination and pose conditions. There are some recent studies [4, 97] showing that recognition accuracies of face recognition systems are strongly affected by extrinsic factors, e.g. head pose [85], lighting condition [10], and intrinsic factors, e.g. facial aging [119], facial expressions. However, according to the

surveys in [4, 97, 127], there is no quantified approach to analyze relationships among these factors in order to decompose the subject factor from other factors so that it can be used effectively in face recognition engines.

One of the leading multilinear analysis approaches is Multilinear Principal Component Analysis (MPCA), or Tensorfaces [108, 110]. This method was based on multilinear algebra to present relationships among factors of given data. Figure 1.3 shows an example of multifactor representation on the Extended Yale-B database. There are three subjects, each with nine poses and 64 lighting conditions chosen in this example. Figure (A) presents the distribution of the first two principal components. Meanwhile figure (B) shows the distribution of 64 lighting conditions of the first subject. Finally, figure (C) shows the facial images of the first subject across 11 lighting conditions and nine different poses.

The heart of Multilinear Principal Component Analysis is to use Principal Component Analysis (PCA) [105] and Higher-order Singular Value Decomposition (HOSVD) [62] in order to decompose a given tensor. Sun et al. [100] presented the High Order Orthogo-

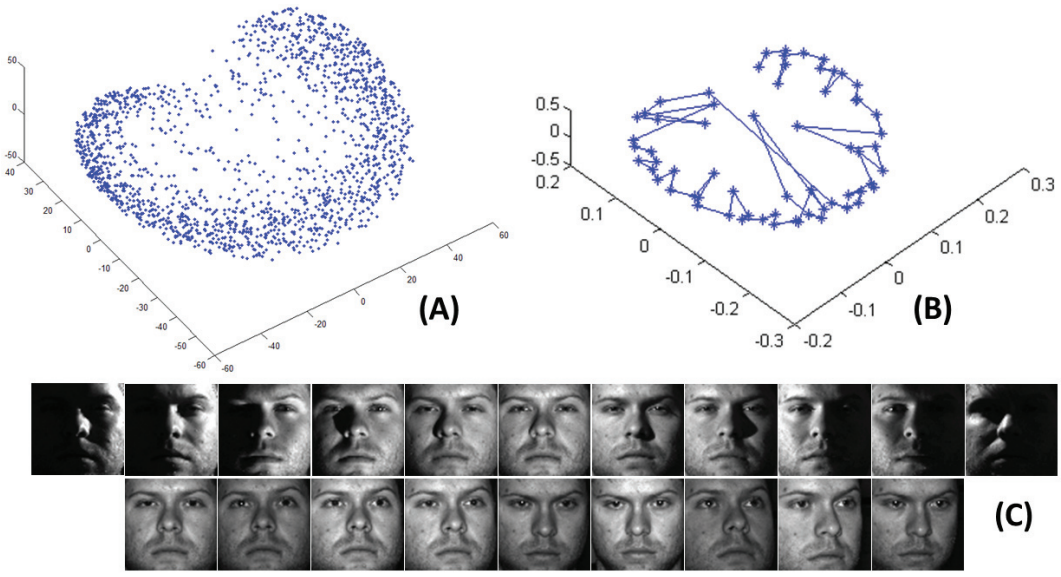


Figure 1.3: Multi-factors in the Extended Yale-B DB: (A) Distribution of the first two principal components trained on first three subjects with nine poses and 64 lighting conditions, (B) Distribution of 64 lighting conditions of the first subject, (C) Facial images of the first subject across 11 lighting conditions and nine different poses.

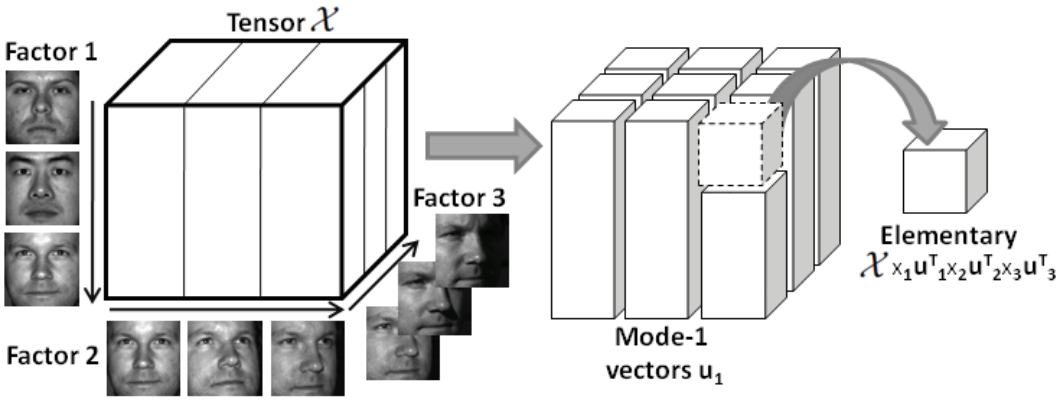


Figure 1.4: Multifactor data presented in Tensor form (left) and the corresponding elementary Tensor projection (right).

nal Iteration (HOOI) to generalize the ideas of Higher-order SVD. Generalized Low Rank Approximation (GLRAM) [64, 123] proceeds the alternative projection to find the optimal projection matrices. Recently, instead of dealing with multilinear approaches, researchers have proposed nonlinear geometrical structures created by multiple factors. Vasilescu and Terzopoulos [108, 109] presented a kernel based MPCA to analyze these nonlinear structures. To fit the manifold structures, created by the variations of body posture and viewpoint in the motion image space, Park and Savvides solved the multifactor analysis using manifold learning algorithms [79]. Pang et al. [77] presented an ℓ_1-norm tensor to solve outliers but their method can easily fail in a local minimum.

1.3 Thesis Contributions

Although widely used, Multilinear Principal Component Analysis still suffers from three major drawbacks. Firstly, it is known that MPCA cannot work on data with missing values, as shown in Figure 1.5. It is also unable to perform well on noisy data or data with outliers. Secondly, since MPCA deals with high multi-dimensional datasets, it is usually computationally expensive. Therefore, it is hard to employ it in practical applications. Finally, MPCA normally loses the original local geometry structures due to the averaging process. Park and Savvides [78, 80] detailed this limitation and presented a Submanifold Preserving

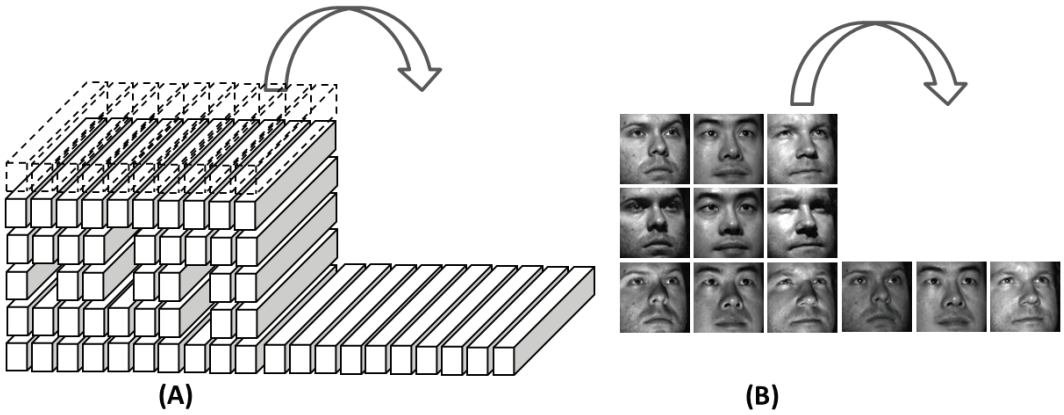


Figure 1.5: (A) Tensor with missing values and (B) the tensor and multifactor flattening process

Multifactor Analysis (SPMA) to keep the factor dependent geometry. The submanifold coordinate is aligned using Procrustes analysis [45] and employs the mean shape as the reference. However, their method is unable to deal with missing values and doesn't allow for high accuracy in local alignment.

This thesis presents a novel approach named Compressed Submanifold Multifactor Analysis (CSMA) to solve the three mentioned problems. Firstly, instead of using the traditional Singular Value Decomposition (SVD), which is unable to deal with input data containing missing values and outliers, a novel Singular Value Decomposition solving via ℓ_1-norm (SVD-ℓ_1) multifactor approach is proposed to decompose factors in given tensors. Our proposed approach can therefore avoid the distortion of outliers efficiently. Secondly, Random Projections (RP) are employed to reduce the number of dimensions of input data in order to reduce the computational time. The theories behind Random Projection in matrix decomposition are also provided to guarantee that it is able to preserve the properties of the compressed multifactor data. Finally, in order to avoid distortions of multifactor structures, a robust local alignment approach is employed to prevent the averaging process.

In addition, this thesis also proposes a novel approach named Sparse Class-dependent Feature Analysis (SCFA), to combine the advantages of sparse representation in an over-complete dictionary, with a powerful nonlinear classifier. The classifier is based on the

estimation of class-specific optimal filters, by solving an ℓ_1-norm optimization problem. We show how this problem is solved using the Alternating Direction Method of Multipliers (ADMM) and also explore relevant convergence details. Our method as well as its Reproducing Kernel Hilbert Space (RKHS) version is tolerant to the presence of noise and other variations in the image. This method achieves very high classification accuracies when applied to the problems of face recognition and object classification.

1.4 Thesis Organization

In chapter 1, we present the motivation for our work, a brief introduction of multiple factor analysis, and our main contributions in this thesis. The remainder of this thesis is organized as follows. Chapter 2 provides background and reviews previous multifactor analysis and tensor methods. In chapter 3, we revisit the area of Compressed Sensing, one of the hot topics in applied mathematics and computer sciences nowadays. The null-space condition, uniqueness and restricted isometry property (RIP) are discussed in detail. In addition, the ℓ_p-norm minimization theories that are of great importance to this thesis are also reviewed carefully. In chapter 4, we present our novel Compressed Submanifold Multifactor Analysis approach that is able to deal with multifactor datasets containing noisy and missing values. Our approach allows the representation of data in a compressed form, while still preserving data structures. We present our SVD-ℓ_1 multifactor decomposition approach to deal with multifactor datasets that contain missing data and outliers. The method borrowed from recent state-of-the-art ideas in ℓ_1-norm formulation is then solved by using Alternative Direction Method of Multipliers optimization method. In chapter 5, we present our novel pattern classification approach, named Sparse Class-dependent Feature Analysis, to combine the advantages of sparse representation in an overcomplete dictionary, with a powerful nonlinear classifier. In order to evaluate our proposed methods, a number of experiments are conducted and are described in detail in chapter 6. In these experiments, our

methods show improvement in both efficiency in dealing with missing data as well as in classification results. Finally, we provide some conclusions and scope for future work in chapter 7.

1.5 Notation

In this thesis, boldface lowercase letters represent vectors, e.g. $\mathbf{x}$, and boldface uppercase letters denote matrices, e.g. $\mathbf{X}$. Higher-order tensors or multidimensional data are denoted by calligraphic uppercase letters, e.g. $\mathcal{X}$. Given a matrix $\mathbf{X}$, $\mathbf{X}^\top$ is the transpose of $\mathbf{X}$. Meanwhile, ℓ_2 denotes the ℓ_2-norm of a vector, i.e. $\forall \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_2 = (\sum_{i=1}^n x_i^2)^{1/2}$, ℓ_1 denotes the ℓ_1-norm of a vector, i.e. $\forall \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$. In a number of studies, it is shown that ℓ_1 norm gives much sparser solution than ℓ_2 norm [30, 35]. The trace-norm of a given matrix $\mathbf{X}$, denoted by $\|\mathbf{X}\|_*$, is computed by the sum of the singular values of $\mathbf{X}$. Finally, $< \mathbf{X}, \mathbf{Y} >$ denotes the trace of $\mathbf{X}^\top \mathbf{Y}$.

Chapter 2

Literature Review

One of the first studies related to tensors and multifactor analysis was carried out by Hitchcock in 1927 [52]. In his work, a tensor was presented as a combination of a specific number of *rank-one* tensors. Cattell [23] proposed new concepts to analyze multiple axes and parallel proportional representations in 1944. Based on these concepts, Carroll et al. [22] developed a method named Canonical Decomposition (CANDECOMP) in 1970 that was widely used among the researchers in the community of psychometrics in 1970. During this period, another popular method for tensor decomposition, named PARAFAC [48], was also introduced by Harshman. CANDECOMP and PARAFAC are the state-of-the-art tensor decomposition methods that have been applied successfully in numerous areas, especially in the field of brain imaging where they were called the topographic components model [48]. In 1963, another well-known method, Tucker decomposition, was introduced [103]. During its development, the method has been called by many different names, such as three-mode PCA [60], N-mode PCA [56], Higher-order SVD [62], N-mode SVD [110] and three-mode factor analysis [103]. This method has been applied successfully in a number of fields, i.e. computer vision, data mining, graph analysis, signal processing, numerical analysis, numerical linear algebra, neuroscience and especially psychometrics and chemometrics [58, 67]. While PARAFAC and the Tucker decomposition methods are fruitful

for certain dense and structured tensors, they are still limited when applied to large-scale and sparse tensors. Hence, Savas and Elden presented Krylov-type methods for tensor decomposition and low-rank approximations in large-scale and sparse data [92]. Several Krylov-type procedures have been subsequently introduced that generalize matrix Krylov methods for tensor computations. The words "multifactor" and "tensor" are used interchangeably with the same meaning in this thesis. A detailed review of tensor methods can be found in [58, 67].

The rest of this chapter is organized as follows. First, we review the preliminaries of multifactor analysis and higher-order tensor decompositions. We then review two main fundamental tensor decomposition methods, i.e. PARAFAC and Tucker decomposition before showing the limitations of these fundamental tensor decomposition methods.

2.1 Mutifactor Analysis Preliminaries

A tensor or a multifactor model can be represented as a multi-dimensional or N-way array, i.e. $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$. In other words, as defined in [58], an Nth-order tensor is a result of the *tensor product* of N vector spaces defined in their own coordinate system. Figure 2.1 shows an example of a *third-order* tensor ($N = 3$). Noticeably, when $N > 3$, the visual representation of that higher-order tensor will become more complicated. Decompositions

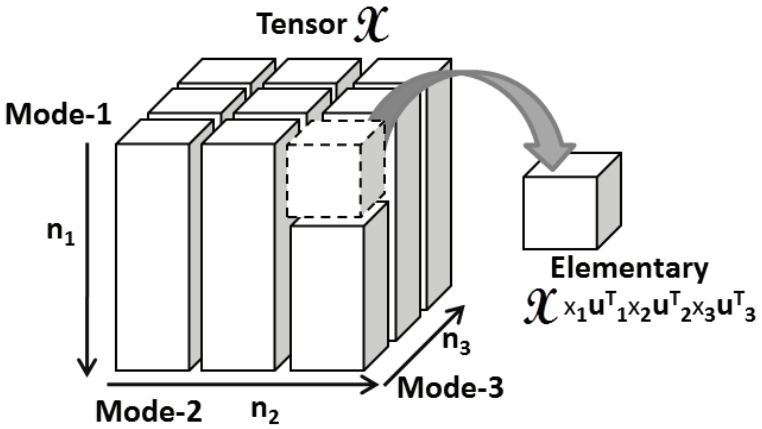


Figure 2.1: An example of a $third\text{-}order$ tensor ($N = 3$), $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

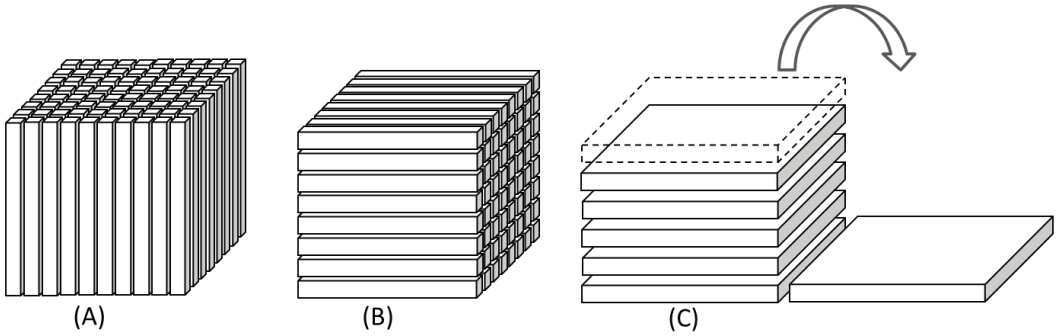


Figure 2.2: Examples of tensor fibers and tensor slices: (A) Mode-1 tensor fibers $x_{:,j,k}$, (B) Mode-2 tensor fibers $x_{i,:,k}$, (C) Horizontal slices $\mathbf{X}_{i,:,:}$

of higher-order tensors have become one of the interesting topics among applied mathematicians for decades [3]. Compared to matrices, higher-order tensors have a couple of differences and are more complicated in definition and representation. In this section, we review the preliminaries of tensors and provide details on the associated notation used in this thesis.

2.1.1 Tensor Terminologies

Tensor Modes

The modes of a given tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$ are the number of different dimensions N of that tensor. They are also called "orders" or "ways" in a number of published articles. Using this definition, matrices can be simply considered as tensors with a mode of two. Higher-order tensors, i.e. third-order or higher, are denoted by boldface Euler script letters as defined in section 1.5. The element (i_1, i_2, i_3) of a third-order tensor $\mathcal{X}$ is denoted by $\mathbf{x}_{i_1,i_2,i_3}$. Figure 2.1 shows an example of a tensor with a mode of three ($N = 3$). Most of tensors defined in this thesis are denoted in a restricted sense, i.e. a three-dimensional array of real values, $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where the vector space is equipped with some algebraic structures to be defined.

Tensor Fibers

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, its tensor fibers can be computed by keeping its all tensor indices except one, i.e. $\mathcal{X}_{:,i_2,i_3,\ldots,i_N}$, $\mathcal{X}_{i_1,:,i_3,\ldots,i_N}$, ..., $\mathcal{X}_{i_1,i_2,\ldots,i_{N-1},:}$.Figures 2.2 (A) and (B) show two examples of mode-1 and mode-2 tensor fibers of a third-order tensor. When computed from a given tensor, fibers are always considered as column vectors.

Tensor Slices

Tensor slices have almost similar properties as tensor fibers except that they involve releasing two factors instead of one. Given an N-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, its tensor slices can be computed by keeping its all tensor indices except two, i.e. $\mathcal{X}_{:,:,i_3,\ldots,i_N}$, $\mathcal{X}_{i_1,:,:,i_4,\ldots,i_N}$, ..., $\mathcal{X}_{i_1,i_2,\ldots,i_{N-2},:,:}$. Figure 2.2 (C) shows the horizontal tensor slides of a third-order tensor $\mathcal{X}$, denoted by $\mathbf{X}_{i_1,:,:}$. Alternatively, the i_3-th frontal slice of a third-order tensor, $\mathbf{X}_{:,:,i_3}$, may be denoted more compactly as $\mathbf{X}_{i_3}$.

2.1.2 Tensor Inner Product and Tensor Norm

Tensor Inner Product

Given two higher-order tensors $\mathcal{X}$ and $\mathcal{Y}$ with the same dimensions, i.e. $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, the *tensor inner product* is computed as in Eqn. (2.1).

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1}^{n_1} \sum_{i_2}^{n_2} \ldots \sum_{i_N}^{n_N} \mathbf{x}_{i_1,i_2,\ldots,i_N} \mathbf{y}_{i_1,i_2,\ldots,i_N} \tag{2.1}$$

Tensor Norm

Based on the definition of tensor inner product, the *tensor norm* or the *Frobenius norm* of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$ is simply defined as in Eqn. (2.2).

$$\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle^{1/2} = (\sum_{i_1}^{n_1} \sum_{i_2}^{n_2} \ldots \sum_{i_N}^{n_N} \mathbf{x}_{i_1,i_2,\ldots,i_N} \mathbf{x}_{i_1,i_2,\ldots,i_N})^{1/2} \tag{2.2}$$

Similar to the property in matrices, multilinear multiplication by orthogonal matrices does not change the Euclidean length of the corresponding fibres of the tensor. Therefore, the tensor norm is invariant to any orthogonal transformation. For example, given a set of orthogonal matrices, i.e. $\mathbf{U}_1$, $\mathbf{U}_2$, $\mathbf{U}_3$, $\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$, the following property in tensors has been proven:

$$\|\mathcal{X}\|_F = \|(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3).\mathcal{X}\|_F = \|\mathcal{X}.(\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3)\|_F \tag{2.3}$$

2.1.3 Rank-One Tensor

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, it is called *rank-one* tensor if and only if it can be represented as the outer product of N vectors as in Eqn. (2.4).

$$\mathcal{X} \triangleq \mathbf{v}^{(1)} \circ \mathbf{v}^{(2)} \circ \mathbf{v}^{(3)} \tag{2.4}$$

where "$\circ$"denotes the outer product of vectors. In other words, each element of the tensor $\mathcal{X}$ is the product of the corresponding vector elements, as defined below:

$$\mathbf{x}_{i_1,i_2,\ldots,i_N} \triangleq \mathbf{v}^{(1)}_{i_1} \mathbf{v}^{(2)}_{i_2} \ldots \mathbf{v}^{(N)}_{i_N}, \forall i_k \in [1, n_k] \tag{2.5}$$

Figure 2.3 shows an example of a third-order tensor that satisfies rank-one property.

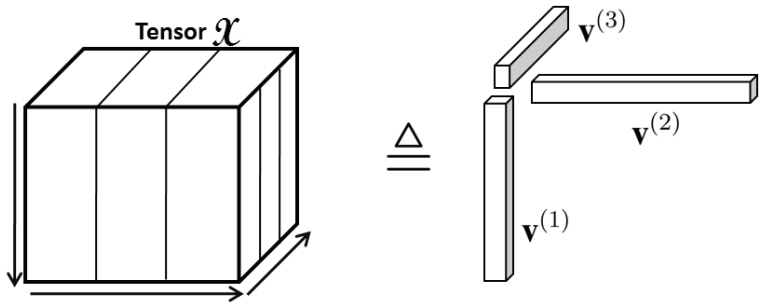


Figure 2.3: An example of tensor representation: the third-order tensor $\mathcal{X}$ is presented under the outer product of three vectors $\mathbf{v}^{(1)}$, $\mathbf{v}^{(2)}$ and $\mathbf{v}^{(3)}$.

2.1.4 Tensor Rank

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, its *rank*, denoted by $rank(\mathcal{X})$, is defined as the smallest number of rank-one tensors whose sum can generate the tensor $\mathcal{X}$. Rank computation in tensors are much more complicated than the one in matrices. Given a random tensor, there is no straightforward method to determine the rank of that tensor since it is an NP-hard problem [49]. In practice, in order to determine the rank of a given tensor, it is usual to numerically fit various rank-K components, as is done in the PARAFAC method, discussed in section 2.2.

2.1.5 Tensor Flattening

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$, its flattening, also called tensor unfolding or tensor matricization, is a technique to reorder the elements of its Nth-order tensor into a matrix. It is clear that the tensor $\mathcal{X}$ can be flattened along its different modes. Details on tensor flattening methods can be found in Kolda et al. [58]. In our work, we are only interested in mode-n flattening. The mode-n flattening of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_N}$ is denoted by $\mathbf{X}_{(n)}$ that arranges the mode-n fibers to be the columns of the resulting matrix. Figure 2.4 shows an example of how a $2 \times 3 \times 2$ tensor can be flattened into three mode-3 unfolding matrices, i.e. $\mathbf{X}_{(1)}$, $\mathbf{X}_{(2)}$ and $\mathbf{X}_{(3)}$.

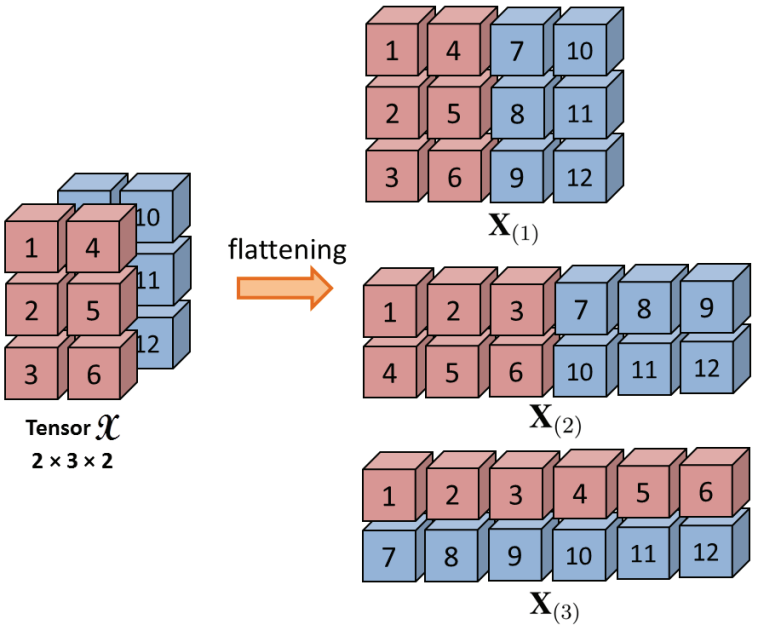


Figure 2.4: An example of tensor flattening. Given a third-order tensor $\mathcal{X} \in \mathbb{R}^{2\times3\times2}$, it can be flattened in mode-3 into three unfolding matrices $\mathbf{X}_{(1)} \in \mathbb{R}^{2\times3\times2}$, $\mathbf{X}_{(2)}$, and $\mathbf{X}_{(3)}$.

2.1.6 n-Mode Product

The n-mode matrix product denotes the multiplication of a tensor by a matrix or a vector in mode n. Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times\ldots\times n_N}$ and a matrix $\mathbf{Y} \in \mathbb{R}^{m\times n_k}$, their k-mode product, which is denoted by $(\mathcal{X} \times_k \mathbf{Y})$, can be computed as follows:

$$(\mathcal{X} \times_k \mathbf{Y})_{i_1,i_2,\ldots,i_{k-1},j,i_{k+1},\ldots,i_N} = \sum_{i_k=1}^{n_k} \mathbf{x}_{i_1,i_2,\ldots,i_N}\mathbf{y}_{j,i_k} \tag{2.6}$$

It is to be noted that in our work, we only consider the multiplication of a tensor by a matrix. However, tensors can be also multiplied together. In that case, their notation and computation will be more complicated. [58] provides details on tensor multiplication.

2.1.7 Matrix Kronecker, Khatri-Rao and Hadamard Products

In this section, we review several important types of products, including the Kronecker product, the Khatri-Rao product, and the Hadamard product. These matrix product methods are used widely in our thesis.

Kronecker Product

Given two matrices $\mathbf{X} \in \mathbb{R}^{m_1 \times n_1}$ and $\mathbf{Y} \in \mathbb{R}^{m_2 \times n_2}$, their *Kronecker product* $\mathbf{K} \in \mathbb{R}^{(m_1 m_2) \times (n_1 n_2)}$, which is denoted by $\mathbf{X} \otimes \mathbf{Y}$, can be computed as follows:

$$\mathbf{K} = \mathbf{X} \otimes \mathbf{Y} = \begin{pmatrix} x_{1,1}\mathbf{Y} & x_{1,2}\mathbf{Y} & ... & x_{1,n_1}\mathbf{Y} \\ x_{2,1}\mathbf{Y} & x_{2,2}\mathbf{Y} & ... & x_{2,n_1}\mathbf{Y} \\ ... & ... & ... & ... \\ x_{m_1,1}\mathbf{Y} & x_{m_1,2}\mathbf{Y} & ... & x_{m_1,n_1}\mathbf{Y} \end{pmatrix}_{(m_1 m_2) \times (n_1 n_2)} \tag{2.7}$$

Khatri-Rao Product

Given two matrices $\mathbf{X} \in \mathbb{R}^{m_1 \times n}$ and $\mathbf{Y} \in \mathbb{R}^{m_2 \times n}$, their *Khatri-Rao product* $\mathbf{R} \in \mathbb{R}^{(m_1 m_2) \times n}$, which is denoted by $\mathbf{X} \odot \mathbf{Y}$, can be computed as follows:

$$\mathbf{R} = \mathbf{X} \odot \mathbf{Y} = [\mathbf{x}_1 \otimes \mathbf{y}_1, \mathbf{x}_2 \otimes \mathbf{y}_2, ..., \mathbf{x}_n \otimes \mathbf{y}_n] \tag{2.8}$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ are columns of $\mathbf{X}$ and $\mathbf{Y}$ respectively. In other words, the Khatri-Rao product can be considered as the Kronecker product of the matching columns of $\mathbf{X}$ and $\mathbf{Y}$. When $\mathbf{x}$ and $\mathbf{y}$ are vectors, the Kronecker and Khatri-Rao products are identical.

Hadamard Product

Given two matrices $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{m \times n}$, their *Hadamard product* $\mathbf{H} \in \mathbb{R}^{m \times n}$, which is denoted by $\mathbf{X} * \mathbf{Y}$, can be computed as follows:

$$\mathbf{H} = \mathbf{X} * \mathbf{Y} = \begin{pmatrix} x_{1,1}y_{1,1} & x_{1,2}y_{1,2} & ... & x_{1,n}y_{1,n} \\ x_{2,1}y_{2,1} & x_{2,2}y_{2,2} & ... & x_{2,n}y_{2,n} \\ ... & ... & ... & ... \\ x_{m,1}y_{m,1} & x_{m,2}y_{m,2} & ... & x_{m,n}y_{m,n} \end{pmatrix}_{m \times n} \tag{2.9}$$

Again, [58] provides more details on these products and their useful properties.

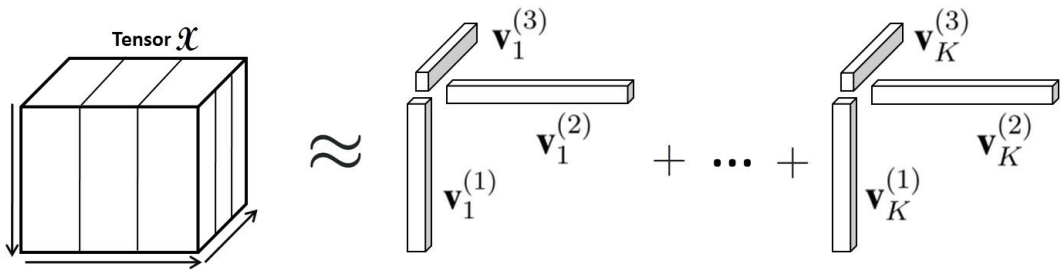


Figure 2.5: An example of PARAFAC tensor decomposition, a tensor $\mathcal{X}$ is approximated by a sum of K components of rank-one tensors.

2.2 PARAFAC Decomposition

The PARAFAC decomposition method factorizes a tensor into a sum of component rank-one tensors. Given a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a redefined positive integer K, the PARAFAC decomposition can be calculated as follows:

$$\mathcal{X} \simeq \sum_{r=1}^{K} \mathbf{x}_r^{(1)} \circ \mathbf{x}_r^{(2)} \circ \mathbf{x}_r^{(3)} \tag{2.10}$$

where $\mathbf{x}_r^{(1)} \in \mathbb{R}^{n_1}$, $\mathbf{x}_r^{(2)} \in \mathbb{R}^{n_2}$ and $\mathbf{x}_r^{(3)} \in \mathbb{R}^{n_3}, r = 1, ..., K$. When denoted in element-wise form, Eqn. (2.10) can be rewritten as follows:

$$\mathbf{x}_{i,j,k} \simeq \sum_{r=1}^{K} \mathbf{x}_{i,r}^{(1)} \mathbf{x}_{j,r}^{(2)} \mathbf{x}_{k,r}^{(3)}, \forall i = 1, ..., n_1; j = 1, ..., n_2; k = 1, ..., n_3. \tag{2.11}$$

Generally, the PARAFAC decomposition of an Nth-order tensor $\mathcal{X}$ can be simply found using the Alternative Least Square (ALS) method as follows:

$$\min_{\widetilde{\mathcal{X}}} \|\mathcal{X} - \widetilde{\mathcal{X}}\| \tag{2.12}$$

where $\widetilde{\mathcal{X}}$ is calculated as given in Eqn. (2.10):

$$\widetilde{\mathcal{X}} = \sum_{r=1}^{K} \lambda_r \mathbf{x}_r^{(1)} \circ ... \circ \mathbf{x}_r^{(N)} = [\lambda; \mathbf{X}^{(1)}, \mathbf{X}^{(2)}, ..., \mathbf{X}^{(N)}] \tag{2.13}$$

Algorithm 1 PARAFAC Decomposition Method [58]

Input: Tensor $\mathcal{X}$, number K
Output: λ, $\mathbf{X}^{(i)}, \forall i = 1, ..., N$
Initialize $\mathbf{X}^{(i)} \in \mathbb{R}^{n_i \times K}, \forall i = 1, ..., N$
repeat
  for $\forall\, n \in [1..N]$ **do**
    $\mathbf{V} \leftarrow \mathbf{X}^{(1)\top}\mathbf{X}^{(1)} * ... * \mathbf{X}^{(n-1)\top}\mathbf{X}^{(n-1)} * \mathbf{X}^{(n+1)\top}\mathbf{X}^{(n+1)} * ... * \mathbf{X}^{(N)\top}\mathbf{X}^{(N)}$
    $\mathbf{X}^{(n)} \leftarrow \mathbf{A}^{(n)}(\mathbf{X}^{(N)} \odot ... \odot \mathbf{X}^{(n+1)} \odot \mathbf{X}^{(n-1)} \odot ... \odot \mathbf{X}^{(1)})\mathbf{V}^{\dagger}$
    normalize columns of $\mathbf{X}^{(n)}$, norms are named as λ
  end for
until fit ceases to improve or maximum iterations exhausted

In order to find $\mathbf{X}^{(k)}$, the Alternative Least Square approach fixes all $\mathbf{X}^{(i)}, \forall i \neq k$. The procedure is repeated until some convergence criterion is satisfied. Assuming we want to solve for $\mathbf{X}^{(1)}$, then the minimization problem can be defined as follows,

$$\min_{\widetilde{\mathbf{X}}^{(1)}} \|\mathbf{X}_{(1)} - \widetilde{\mathbf{X}}^{(1)}(\mathbf{X}^{(N)} \odot \mathbf{X}^{(N-1)} \odot ... \odot \mathbf{X}^{(2)})^{\top}\|_F \tag{2.14}$$

where $\widetilde{\mathbf{X}}^{(1)} = \mathbf{X}^{(1)}.\,\mathrm{diag}(\lambda)$. Then, we can find the optimal solution to the problem (2.14) as follows:

$$\widetilde{\mathbf{X}}^{(1)} = \mathbf{X}_{(1)}[(\mathbf{X}^{(N)} \odot \mathbf{X}^{(N-1)} \odot ... \odot \mathbf{X}^{(2)})^{\top}]^{\dagger} \tag{2.15}$$

Due to the pseudoinverse property of the Khatri-Rao product, Eqn. (2.15) can be written as follows:

$$\widetilde{\mathbf{X}}^{(1)} = \mathbf{X}_{(1)}(\mathbf{X}^{(N)} \odot ... \odot \mathbf{X}^{(2)})(\mathbf{X}^{(N)\top}\mathbf{X}^{(N)} * ... * \mathbf{X}^{(2)\top}\mathbf{X}^{(2)})^{\dagger} \tag{2.16}$$

Algorithm (1) shows the pseudocode of the PARAFAC decomposition algorithm. This method is simple to understand and implement. However, it is easy to see that there are two limitations to this method. Firstly, it is assumed that the number of component rank-one tensors K has to be given. Secondly, The solution is not guaranteed to converge to a global minimum. In addition, the final solution is also heavily dependent on the starting guess. Figure 2.5 shows an example of how the FARAFAC tensor decomposition method works.

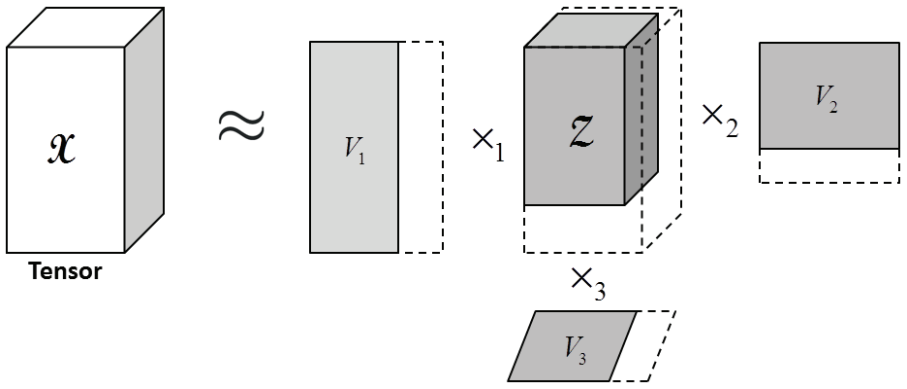


Figure 2.6: An example of Tucker decomposition, a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is approximated by a combination of a core tensor $\mathcal{Z} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ and three factor matrices, i.e. $\mathbf{V}_1 \in \mathbb{R}^{n_1 \times m_1}$, $\mathbf{V}_2 \in \mathbb{R}^{n_2 \times m_2}$ and $\mathbf{V}_3 \in \mathbb{R}^{n_3 \times m_3}$.

Algorithm 2 HOSVD Decomposition Method [58, 62]

Input: Tensor $\mathcal{X}$
Output: $\mathcal{Z}$, $\mathbf{X}^{(i)}$, $\forall i = 1, ..., N$
for $\forall\ n \in [1..N]$ **do**
  $\mathbf{X}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{X}_{(n)}$
end for
$\mathcal{Z} \leftarrow \mathcal{X} \times_1 \mathbf{X}^{(1)\top} \times_2 \mathbf{X}^{(2)\top} ... \times_N \mathbf{X}^{(N)\top}$,

2.3 Tucker Decomposition

Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_N}$, the Tucker decomposition factorizes it into a core tensor $\mathcal{Z} \in \mathbb{R}^{m_1 \times m_2 \times ... \times m_N}$ multiplied by a matrix $\mathbf{V}_i \in \mathbb{R}^{n_i \times m_i}$ along each mode i. Mathematically, the Tucker decomposition can be represented as follows:

$$\mathcal{X} = \mathcal{Z} \times_1 \mathbf{V}_1 \times_2 \mathbf{V}_2 \times_3 ... \times_N \mathbf{V}_N = \sum_{i_1,...,i_N}^{n_1,...,n_N} z_{i_1,...,i_N} \mathbf{V}_{1,i_1,...,i_N} \circ ... \circ \mathbf{V}_{N,i_1,...,i_N} \quad (2.17)$$

Notice that these factor matrices are usually orthogonal. The Tucker decomposition became more popular after the publication of the Higher-order SVD method, that was proposed by Lathauwer [62]. The HOSVD decomposition method is summarized by Algorithm (2).

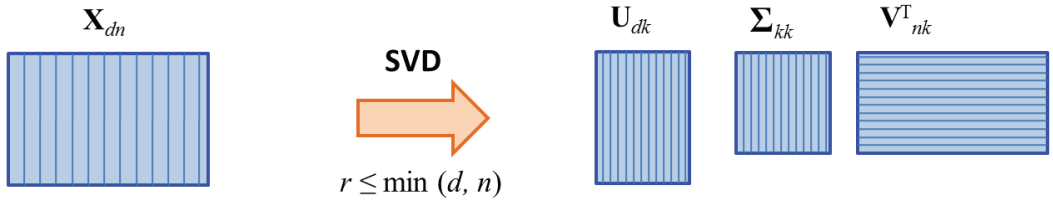


Figure 2.7: An example to show the decomposition process of the classical Singular Value Decomposition method. Given a matrix **A** of size $d \times n$, the SVD method will decompose **A** into two orthonormal matrices **U** and **V**, and a diagonal matrix Σ.

2.4 Higher-order SVD (HOSVD)

Multilinear PCA is an extension of Principal Component Analysis to multi-factor frameworks, where SVD is at the heart of the decomposition process. This section first reviews the traditional computation of SVD and then discusses its limitations.

2.4.1 Singular Value Decomposition (SVD)

Given n training images, each with d pixels, denoted by a 2D matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, a pair of *singular vectors* $\mathbf{u} \in \mathbb{R}^d$ and $\mathbf{v} \in \mathbb{R}^n$ of $\mathbf{X}$ can be computed using Eqn. (2.18).

$$\mathbf{X}\mathbf{v} = \lambda\mathbf{u} \quad \text{and} \quad \mathbf{u}^\top\mathbf{X} = \lambda\mathbf{v}^\top \tag{2.18}$$

where $\lambda \in \mathbb{R}$ is the corresponding *singular value*. Generally, $\mathbf{X}$ can be reformulated [55] as shown below:

$$\mathbf{X} = \sum_{i=1}^{r} \lambda_i \mathbf{u}_i \mathbf{v}_i^\top \quad \text{or} \quad \mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \tag{2.19}$$

where r denotes the rank of $\mathbf{X}$, $r \leq \min(d, n)$ and $\mathbf{u}_i \in \mathbb{R}^d$ and $\mathbf{v}_i \in \mathbb{R}^n$ are orthonormal, where each has the length of 1 and every pair is orthogonal, i.e. $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$. $\mathbf{\Sigma}$ is a diagonal matrix containing the square root of the eigenvalues of $\mathbf{U}$ or $\mathbf{V}$ in descending order. Each ($\mathbf{u}_i$, $\mathbf{v}_i$) pair forms a pair of left and right singular vectors with singular value $\lambda_i > 0$, where $\lambda_k \geq \lambda_{k+1}, \forall k \in [1, r-1]$. It follows that each $\mathbf{u}_i$ is an eigenvector of $\mathbf{X}\mathbf{X}^\top$ and each $\mathbf{v}_i$ is an eigenvector of $\mathbf{X}^\top\mathbf{X}$, and the corresponding eigenvalues are λ_i^2. In

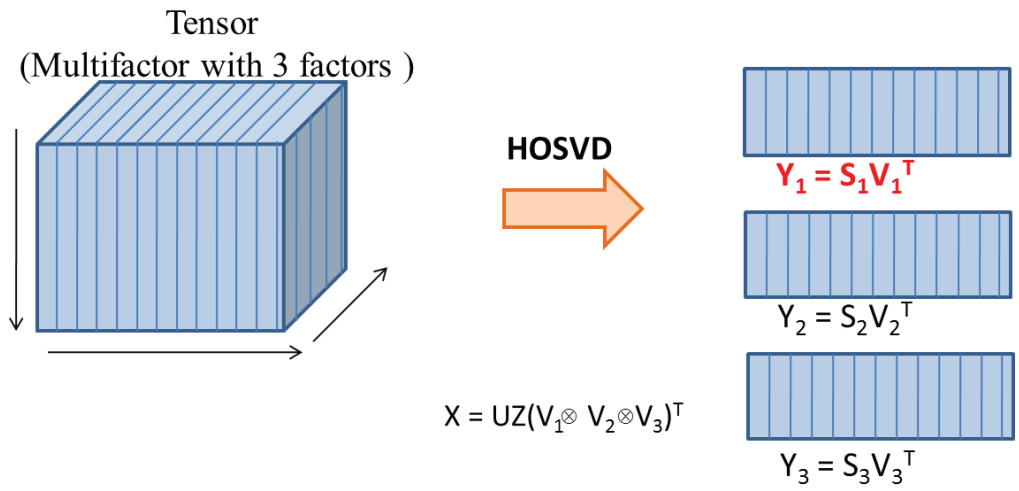


Figure 2.8: An example to show the decomposition process of the Higher-order SVD method. Given a third-order tensor $\mathcal{X}$, HOSVD method will decompose it into three matrices $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$, the core tensor $\mathcal{Z}$ and the matrix $\mathbf{U}$ that satisfy the HOSVD condition.

order to find the top singular vectors, the unit vector $\mathbf{v}_1$ that maximizes $\|\mathbf{X}\mathbf{v}\|$ will be first computed and then $\mathbf{u}_1$ is found from that. Generally, to compute the complete SVD, we first find $\mathbf{u}_1, \mathbf{v}_1$ and λ_1. Then we iteratively employ this on the matrix $(\mathbf{X} - \lambda_1 \mathbf{u}_1 \mathbf{v}_1^\top)$. In other words, a rank-1 matrix is subtracted at each iteration.

2.4.2 Higher-Order SVD (HOSVD)

Higher-Order SVD [62] is a multilinear generalization of Singular Value Decomposition. Given an Nth-order tensor $\mathcal{X}$, HOSVD can decompose it into a core tensor $\mathcal{Z}$ and N orthogonal matrices, i.e. a matrix $\mathbf{U}$ for pixel values and N matrices $\mathbf{V}_i$ to represent N factors. Without lost of generality, assume that $n = 3$. Thus, a tensor $\mathcal{X} \in \mathbb{R}^{d \times n_1 \times n_2 \times n_3}$ can be decomposed using HOSVD as follows:

$$\mathcal{X} = \mathcal{Z} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}_1^\top \times_3 \mathbf{V}_2^\top \times_4 \mathbf{V}_3^\top \tag{2.20}$$

where $\times_k$ is the *k-mode* matrix product of a tensor, as defined in Section 2.1.6. Figure 2.9 (top) shows an example of the HOSVD decomposition. In Multilinear PCA, HOSVD is

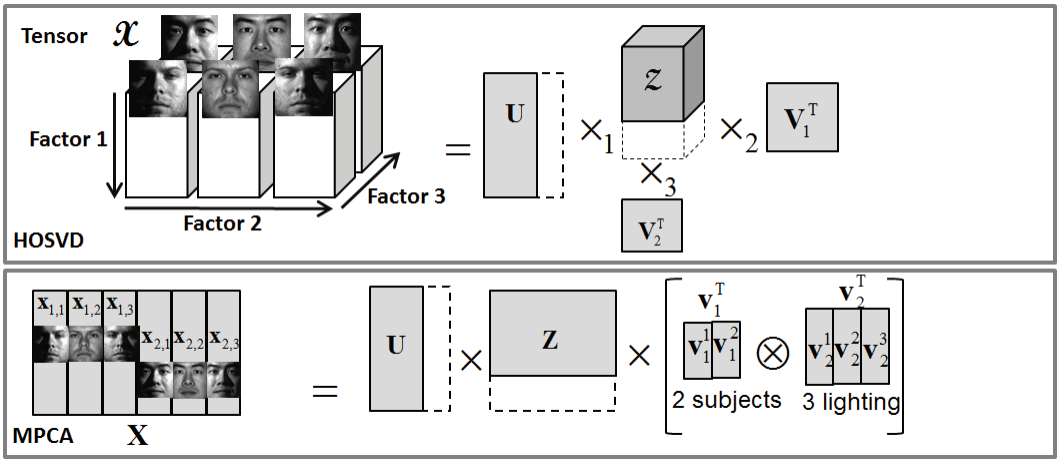


Figure 2.9: Higher-order SVD is a multilinear generalization of the SVD. In HOSVD, the third-order tensor $\mathcal{X}$ is decomposed into one core tensor $\mathcal{Z}$ and three orthogonal matrices: matrix $\mathbf{U}$ (pixels), factor $\mathbf{V}_1$ (subjects), and factor $\mathbf{V}_2$ (lighting). The columns of each orthogonal matrix form the basis of each of the three vector spaces of a tensor $\mathcal{X}$. In MPCA, HOSVD is reformulated in terms of matrices, instead of tensors, using the Kronecker product. In MPCA, the first three images of subject 1 are represented on the first column $\mathbf{V}_1^1$ of $\mathbf{V}_1^\top$, while the next three images of subject 2 depend on the second column $\mathbf{V}_1^2$ of $\mathbf{V}_1^\top$. The same representation is used for 3 lighting conditions $\mathbf{V}_2^\top$. Note that $\mathbf{U}$ is identical to the matrix $\mathbf{U}$ in PCA.

reformulated in terms of matrices, instead of tensors, using the Kronecker product. The equivalent form of Eqn. (2.20) in MPCA can be presented as follows,

$$\mathbf{X} = \mathbf{U}\mathbf{Z}(\mathbf{V}_1 \otimes \mathbf{V}_2 \otimes \mathbf{V}_3)^\top \tag{2.21}$$

where $\otimes$ denotes the Kronecker product (defined in section 2.1.7). $\mathbf{U}$ is identical to the matrix $\mathbf{U}$ in Eqn. (2.19). A matrix $\mathbf{Z}$ results from the pixel-mode flattening of a core tensor presented in [108]. $\mathbf{V}_k$ is the right singular vector matrix of the flatten tensor $\mathbf{X}$ along the factor k. From Eqn. (2.21), $\mathbf{Z}$ can be derived as follows,

$$\mathbf{Z} = \mathbf{U}^\top\mathbf{X}(\mathbf{V}_1 \otimes \mathbf{V}_2 \otimes \mathbf{V}_3) \tag{2.22}$$

Figure 2.9 (bottom) shows how Multilinear PCA decomposition can be carried out.

Chapter 3

Compressed Sensing Revisited

3.1 Sensing Method

The revolutionary fields of compressed sensing and sparse signal approximation have been rapidly developed during this decade [6, 16, 18, 19, 20, 21, 29, 87]. Alongside the accelerated growth in numerous applications, such as: medical image processing [68, 69], single-pixel imaging [32], face recognition systems [113, 114], etc., the theories of sparse and compressible signal representation have been fully enriched by many researchers in this field [16, 19, 87]. Compressed sensing [88, 101] is defined as a signal acquisition paradigm that allows recovering estimates of compressible and sparse signals from high ambient dimensions N using linear measurements M with much fewer dimensions (i.e., $M \ll N$). Therefore, the protocol in compressed sensing aims to directly acquire only important information from a given signal. This approach has the ability to acquire and recover signals in the most efficient way possible and avoid a data deluge.

3.1.1 Motivation

The motivation for compressed sensing can be envisioned by considering the following problem. Given a matrix $\mathbf{\Phi}$ of size $M \times N$, where $M \ll N$, and a signal $\mathbf{y} \in \mathbb{R}^M$, the

question that arises is if there exists a *sparse* vector $\mathbf{x}$ such that $\mathbf{y} \simeq \mathbf{\Phi x}$ or not. In other words, we want to find a sparse vector or a set of sparse coefficients $\mathbf{x}$ so that the signal $\mathbf{y}$ can be approximated by projecting this sparse vector $\mathbf{x}$ onto the given dictionary $\mathbf{\Phi}$. The sparsity in this case is measured by counting the number of non-zero entries in a given vector, i.e., $\|\mathbf{x}\|_{\ell_0}$ = number of non-zero entries in $\mathbf{x}_0 = \sum_i I(x_i \neq 0)$. It is clear that the number of non-zero entries in $\mathbf{x}$ has to be smaller or equal to M at most. The matrix $\mathbf{\Phi}$ is also called a dictionary or overcomplete dictionary or frame in some contexts. From these definitions, the problem can be redefined as a problem of finding the sparsest vector $\mathbf{x}_0$ that satisfies the linear system $\mathbf{y} = \mathbf{\Phi x}$. In the optimization framework, the problem can be denoted as shown in Eqn. (3.1).

$$P(0) : \min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_0} \qquad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi x} \tag{3.1}$$

There are three important questions regarding Eqn. 3.1 that need to be answered that have resulted in three key research topics in compressed sensing [90]. Firstly, when is the solution to $P(0)$, as shown in Eqn. (3.1), unique? Secondly, what are the most efficient ways to solve for $P(0)$? Finally, given a training data set $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N$, how can we learn a dictionary $\mathbf{\Phi}$ so that it can support sparse representations on the training data set $\mathbf{X}$?

There are many variants of $P(0)$ in the definition of the Eqn. (3.1). They are listed in the following equations:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_0} \qquad \text{subject to} \quad \mathbf{y} = \mathbf{\Phi x} \tag{3.2a}$$

$$\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_0} \qquad \text{subject to} \quad \|\mathbf{y} - \mathbf{\Phi x}\| \leq \epsilon \tag{3.2b}$$

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi x}\| \qquad \text{subject to} \quad \|\mathbf{x}\|_{\ell_0} \leq K \tag{3.2c}$$

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{\Phi x}\|^2 + \tau^2 \|\mathbf{x}\|_{\ell_0} \tag{3.2d}$$

where ϵ denotes the reconstruction error permitted by the system, K is the maximum

number of non-zero entries allowed in reconstructed signals, τ is a trade-off parameter between sparsity and reconstruction fidelity. It must be noted that if the support set Ω is given in the solution to $P(0)$, then the problem reduces to that of a simple feasibility problem, i.e. $y \in R(\mathbf{\Phi}_\Omega)$, as in numerous well-known problems such Matching Pursuit (MP) [11, 74], Orthogonal Matching Pursuit (OMP) [83], etc. However, it is very hard to find the support set Ω in practical applications. One approach is to enumerate all possible supports, and then pick the the smallest one that leads to a feasible solution. This method, however, is a combinatorial and exponential time algorithm, since $P(0)$ is reducible to an NP-complete problem. In a general case, this problem cannot be solved in polynomial time. However, a fast solution may exist in some special cases of signals. In this section, we discuss when signals can be recovered.

3.1.2 Null-space Condition

Given a matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, the *null-space* of the matrix $\mathbf{\Phi}$ is defined as follows:

$$N(\mathbf{\Phi}) = \alpha : \mathbf{\Phi}\alpha = \mathbf{0}. \tag{3.3}$$

From this definition, it can be concluded that for any two distinct sparse signals $\mathbf{x}$ and $\mathbf{x}'$ that need to be recovered, where $\mathbf{x}, \mathbf{x}' \in \Sigma_k$, the following constraint holds: $\mathbf{x} \neq \mathbf{x}'$. This is because it is impossible to find two different vectors $\mathbf{x}$ and $\mathbf{x}'$ from the same given measurement $\mathbf{y}$ and the same dictionary $\mathbf{\Phi}$ as in Eqn. (3.1) with the constraint of $\mathbf{x} = \mathbf{x}'$. Assume that $\mathbf{\Phi x} = \mathbf{\Phi x}'$. Then we have $\mathbf{\Phi x} - \mathbf{\Phi x}' = \mathbf{0}$ or $\mathbf{\Phi}(\mathbf{x} - \mathbf{x}') = \mathbf{0}$, where $(\mathbf{x} - \mathbf{x}') \in \Sigma_k$. From this condition, the signal $\mathbf{x}$ can be uniquely recovered from dictionary $\mathbf{\Phi}$ if and only if there doesn't exist any vector in Σ_{2k} in the null-space $N(\mathbf{\Phi})$. In compressed sensing, the *spark* property is one of the most common method used to evaluate this condition. The spark computation will be discussed in detail in section 3.1.3. The fundamental idea in this section is that sparse vectors in the null-space of the matrix have to limit uniqueness. Such

conditions are referred to as null-space conditions.

Given a matrix $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$, assume that these exists a vector $\eta \in N(\mathbf{\Phi})$ such that $\|\eta\|_{\ell_0} \leq 2K$. Let Ω be the support set, $\Omega = supp(\eta)$. Now, we can construct a K-sparse vector $\mathbf{x}$, such that $supp(\mathbf{x}) = \Lambda \subset \Omega$, and $\mathbf{x}_\Lambda = \eta_\Lambda$. Given $\mathbf{y} = \mathbf{\Phi x}$, both $\mathbf{x}$ and $(\mathbf{x} - \eta)$ are in the solution set, furthermore $\|\mathbf{x} - \eta\|_{\ell_0} \leq \|\mathbf{x}\|_{\ell_0}$.

3.1.3 Spark

Definition of Spark [35]. *The spark of a matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_N], \boldsymbol{\phi}_i \in \mathbb{R}^M$ is defined as the* smallest *set of linearly* dependent *columns.*

The definition of $Spark(\mathbf{\Phi})$ is contrast to the one of $Rank(\mathbf{\Phi})$ defined as the largest set of linearly independent columns. From the spark definition, the following condition can be deduced:

$$2 \leq Spark(\mathbf{\Phi}) \leq Rank(\mathbf{\Phi}) + 1 = M + 1 \tag{3.4}$$

In practice, dictionaries are typically full-rank and overcomplete, with $M < N$. Matrices whose entries are independent and identically distributed (i.i.d.) sampled from random distributions typically have $Spark(\mathbf{\Phi}) = M + 1$. Given a random square matrix $\mathbf{\Phi}$ of size $M \times M$, such as Gaussian ones, it is almost surely non-singular.

Lemma (Null-space and Spark Constraint) [35]. *If $\boldsymbol{n} \in N(\mathbf{\Phi})$, then*

$$\|\boldsymbol{n}\|_{\ell_0} \geq Spark(\mathbf{\Phi})$$

Proof. Assume that $\exists \mathbf{n} \in N(\mathbf{\Phi})$, and we have $\|\mathbf{n}\|_{\ell_0} < Spark(\mathbf{\Phi})$.

Then, let $\Omega = supp(\mathbf{n}) = i | \mathbf{n}_i \neq 0$.

It is to be noted that $|\Omega| = \|\mathbf{n}\|_{\ell_0} < Spark(\mathbf{\Phi})$.

Therefore, the columns of $\mathbf{\Phi}$ corresponding to indices in Ω are linearly dependent.

Hence, $Spark(\mathbf{\Phi}) \leq |\Omega| = \|\mathbf{n}\|_{\ell_0} < Spark(\mathbf{\Phi})$. This is a contradiction and therefore the lemma is true. □

3.1.4 Uniqueness via Spark

Theorem (Gorodnitsky-Rao, 1997; Donoho-Elad, 2003). *If a system* $\boldsymbol{y} = \mathbf{\Phi}\boldsymbol{x}$ *has a solution* $\boldsymbol{x}_0$ *such that* $\|\boldsymbol{x}\|_{\ell_0} < Spark(\mathbf{\Phi})/2$*, then* $\boldsymbol{x}_0$ *is also the unique sparsest solution.*

Proof. Recall from Lemma of null-space and spark constraint, that

If $\mathbf{n} \in N(\mathbf{\Phi})$ then $\|\mathbf{n}\|_{\ell_0} \geq Spark(\mathbf{\Phi})$.

Suppose $\exists \mathbf{x}_0, \mathbf{z}_0$ such that $\mathbf{y} = \mathbf{\Phi}\mathbf{x}_0 = \mathbf{\Phi}\mathbf{z}_0$ and $\|\mathbf{z}_0\|_{\ell_0} < \|\mathbf{x}_0\|_{\ell_0}$.

Then, since $\mathbf{\Phi}(\mathbf{z}_0 - \mathbf{x}_0) = \mathbf{0}$, therefore $(\mathbf{z}_0 - \mathbf{x}_0)$ is also in null-space, i.e. $(\mathbf{z}_0 - \mathbf{x}_0) \in N(\mathbf{\Phi})$. From the definition of spark, we have:

$$\|\mathbf{x}_0\|_{\ell_0} + \|\mathbf{z}_0\|_{\ell_0} \geq \|\mathbf{x}_0 + \mathbf{z}_0\|_{\ell_0} \geq spark(\mathbf{\Phi}).$$

This result contradicts clearly the theorem above, where $\|\mathbf{x}\|_{\ell_0} < Spark(\mathbf{\Phi})/2$. Therefore, $\mathbf{x}_0$ is the unique sparsest solution. □

However, spark usually cannot be computed in practice. It is immediately clear that there are C_S^N subset selections needed to be verified either $Spark(\mathbf{\Phi}) \leq S$ or $Spark(\mathbf{\Phi}) \geq S$. The computation of $Spark(\mathbf{\Phi})$ is a combinatorial problem. Its computation is quite expensive. Therefore, we resort to *coherence* as a way to estimate, at least the lower bound, the spark.

3.1.5 Coherence

Definition of Coherence [35]. *Given a matrix* $\mathbf{\Phi} = [\phi_1, \phi_2, ..., \phi_N]$ *with unit-norm columns* ($\|\phi_j\| = 1$)*, its* mutual coherence *is defined as the largest absolute normalized inner product between different columns in* $\mathbf{\Phi}$*. Generally, the mutual coherence* $\mu(\mathbf{\Phi})$ *is denoted as*

follows:

$$\mu(\mathbf{\Phi}) = \max_{1 \leq i,j \leq N, i \neq j} \frac{|\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle|}{\|\boldsymbol{\phi}_i\| . \|\boldsymbol{\phi}_j\|}$$

The mutual-coherence $\mu(\mathbf{\Phi})$ measures how similar two elements of the given dictionary $\mathbf{\Phi}$. In other words, it is a way to evaluate the dependence between columns in $\mathbf{\Phi}$. When the dictionary $\mathbf{\Phi}$ is unitary where all pairwise columns are orthogonal, then mutual-coherence $\mu(\mathbf{\Phi})$ equals zero. In the regular cases of compressed sensing, where the dictionary $\mathbf{\Phi}$ usually has more columns than rows, the mutual-coherence $\mu(\mathbf{\Phi})$ must be positive. In addition, the smaller mutual-coherence $\mu(\mathbf{\Phi})$ is, the better it is to support representations with higher sparsity. Ideally, we always aim to achieve the smallest possible coherence in a given very large dictionary. However, it is usually impossible in practice. The following theorem shows the lower bound of the coherence μ computed from a dictionary $\mathbf{\Phi}$.

Theorem (Welsh, 1974; Strohmer-Heath, 2004). *Given a general $M \times N$ full-rank dictionary $\mathbf{\Phi}$, where $(M \ll N)$, the following condition always holds:*

$$\mu(\mathbf{\Phi}) \geq \sqrt{\frac{N-M}{M(N-1)}} \approx \frac{1}{\sqrt{M}}$$

Proof. This theorem can be proved as follows:

Let $\mathbf{G} = \mathbf{\Phi}^\top \mathbf{\Phi}$, then it is clear that $Rank(\mathbf{G}) = M$,

Let λ_i be the eigenvalues of $\mathbf{G}$, where $i = 1, ..., M$.

Then, we have $\sum_{i=1}^{M} \lambda_i = Trace(\mathbf{G}) = N$. Apply $\ell_2 - \ell_1$ norm equivalence, the following condition is achieved:

$$\|\mathbf{G}\|_F^2 = \sum_i \lambda_i^2 \geq (\sum_i \lambda_i / \sqrt{M})^2$$

or

$$\|\mathbf{G}\|_F^2 \geq \frac{N^2}{M}$$

It is to be noted that $\mathbf{G}_{i,i} = 1$ and $\mu(\mathbf{\Phi}) \geq |\mathbf{G}_{i,j;i\neq j}|$, then

$N + N(N-1)\mu^2(\mathbf{\Phi}) \geq \frac{N^2}{M}$.

Therefore, $\mu^2(\mathbf{\Phi}) \geq \frac{N-M}{M(N-1)}$. □

3.1.6 Uniqueness via Coherence

Theorem [35]. *If a given system* $\boldsymbol{y} = \mathbf{\Phi}\boldsymbol{x}$ *has a solution* $\boldsymbol{x}_0$ *such that* $\|\boldsymbol{x}\|_{\ell_0} < \frac{1}{2}(1 + \frac{1}{\mu(\mathbf{\Phi})})$, *then* $\boldsymbol{x}_0$ *is also the unique sparsest solution.*

The bound of this condition has been determined to be $(1 + 1/\mu(\mathbf{\Phi})) \leq 1 + \sqrt{M}$. Meanwhile, as defined in section 3.1.3, $Spark(\mathbf{\Phi}) \leq M + 1$. Therefore, the difference between $(1+1/\mu(\mathbf{\Phi}))$ and $Spark(\mathbf{\Phi})$ is usually very large. However, it is only the sufficient condition and very pessimistic.

3.2 Measurement Matrices in Compressed Sensing

In compressed sensing, given a linear system defined as follows:

$$\mathbf{y} = \Phi\mathbf{x} + \mathbf{e} \tag{3.5}$$

where $\mathbf{x} \in \mathbb{R}^N$ denotes the unseen signal that the system is trying to recover. The signal $\mathbf{x}$ is usually supposed to be sparse. However, in most practical applications, it will be compressible or approximately sparse in a given transform basis such as wavelet, Hadamard or random projection matrices. Meanwhile, vector $\mathbf{y} \in \mathbb{R}^M$ defines the measurement vector that is usually available in practical applications. $\mathbf{e} \in \mathbb{R}^M$ denotes the measurement noise that is typically modeled as bounded in energy or with a known statistical model such as the Gaussian or Poisson distributions. The nature of these noises usually depends on a

particular application. Finally $\Phi \in \mathbb{R}^{M \times N}$ denotes the *measurement* or the *sensing matrix*. The design of Φ plays a crucial role in the analysis of compressed sensing. In compressed sensing problems, we are usually interested in the *underdetermined* problem where we have fewer equations than unknowns which is clearly ill-posed, i.e. $M < N$. The problem now is to study the possibility of reconstructing $\mathbf{x}$ with high accuracy. Assuming that the observed system doesn't have noise, then the reconstruction signal $\mathbf{x}^*$ can be determined as follows:

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x}\|_{\ell_1} \quad \text{subject to} \quad \Phi\mathbf{x} = \mathbf{y} \tag{3.6}$$

3.2.1 Optimal Measurement Matrices

In practice, most designed sensors are usually linear. Therefore, they are just instantiations of the measurement matrix Φ. For example, the pinhole camera can be modeled as an identity operator, i.e. each pixel directly observes the intensity level of light in a cone, center on the pinhole, in the scene. The resolution of the camera and the camera geometry determines the size of the cone.

When the measurement matrix Φ is square with the sizes of $M \times M$ and *invertible*, from Eqn. (3.5), the estimate of the solution can be found as in Eqn. (3.7).

$$\hat{\mathbf{x}} = \Phi^{-1}\mathbf{y} = \mathbf{x} + \Phi^{-1}\mathbf{e} \tag{3.7}$$

And the error incurred by this estimate can be computed as in Eqn 3.8.

$$\|\mathbf{x} - \hat{\mathbf{x}}\| = \|\Phi^{-1}\mathbf{e}\| \leq \|\Phi^{-1}\|\|\mathbf{e}\| \tag{3.8}$$

Therefore, the key question in this section is that: *what are good measurement matrices?*. Clearly, without any other constraints, we can arbitrarily increase the eigenvalues of Φ, and the error will decrease to zero. Most applications introduce constraints that avoid such naive solutions.

- *Bounded energy:* $\|\Phi\|_F \leq B$: common in applications where sensing is active, such as: radar, sonar, X-ray, etc.
- *Bounded entries:* $\Phi_{i,j} \leq 1$: in most passive sensors, the elements of the signal cannot be amplified.
- *Non-negativity and passive:* $0 \leq \Phi_{i,j} \leq 1$: Light, in incoherent imaging regimes, is non-negative. Hence, most applications involving light is restricted to have measurement matrices that are non-negative.
- *Toeplitz structure:* in many applications, measurement involves convolution of a pulse with the scene structure. In these cases, the measurement matrix is Toeplitz as well as bounded in energy (energy is equal to that of the pulse send out), i.e. radar.

Suppose we look at matrices with $|\Phi_{i,j}| \leq 1$. Then, the error incurred by the estimator $\hat{\mathbf{x}} = \Phi^{-1}\mathbf{y}$ is lower bounded.

$$\|\mathbf{x} - \hat{\mathbf{x}}\| = \|\Phi^{-1}\mathbf{e}\| \geq \|\mathbf{e}\|/\sqrt{M} \tag{3.9}$$

Matrix constructions that achieve the lower bound are referred to as *Hadamard matrices*. But Hadamard matrices are not known for all values of M. There are many constructions. Of these, Sylvester's construction is very famous, as it constructs for $M = 2^Q$ and has a fast transform.

If we restrict ourselves to matrices that are non-negative as well $(0 \leq \Phi_{i,j} \leq 1)$, then for an $M \times M$ matrix, $\|\mathbf{x} - \hat{\mathbf{x}}\| \geq 2\|\mathbf{e}\|/\sqrt{M+1}$. A variant of the Hadamard matrices called *S-matrices* almost achieve this.

In the underdetermined linear system, which is of interest to us. Our focus is solely on the ℓ_1-norm minimization problem as applied to the linear system $\mathbf{y} = \Phi\mathbf{x} + \mathbf{e}$. However, it is impossible to extend the defined problem into non-sparse signals, i.e. $\mathbf{x}$ is not exactly K-sparse, but at best well approximated by a K-sparse signal, in the presence of measurement noise $\mathbf{e}$.

Given $\mathbf{x}^0 \in \mathbb{R}^N$, the K-sparse signal that best approximates $\mathbf{x}^0$ in ℓ_p-norm can be computed as follows:

$$\hat{\mathbf{x}}_K^0 = \arg \min_{\|\mathbf{x}\|_0 \leq K} \|\mathbf{x}^0 - \mathbf{x}\|_p \tag{3.10}$$

A solution can be easily found by selecting the top K-entries in magnitude of $\mathbf{x}_0$ and replacing the other entries by zero. Note that this solution is potentially non-unique when $\mathbf{x}_0$ has entries with equal magnitude. We can define the error in this K-sparse estimate as $\sigma_K(\mathbf{x}^0)_p$ that can be computed as follows:

$$\sigma_K(\mathbf{x}^0)_p = \|\mathbf{x}^0 - \hat{\mathbf{x}}_K^0\|_p = \min_{\|\mathbf{x}\|_0 \leq K} \|\mathbf{x}^0 - \mathbf{x}\|_p \tag{3.11}$$

Note that, this is the least possible error when a non-sparse vector is approximated using a K-sparse vector. This error is associated with a nonlinear estimator that has full access to $\mathbf{x}^0$.

3.2.2 Null-Space Property (NSP)

Definition of NSP. *Given a matrix $\mathbf{\Phi}$, it is said to satisfy the null-space property (NSP) of order K with constant $0 < \gamma < 1$, if $\forall \eta \in N(\mathbf{\Phi})$ and $\forall \Lambda \subset \{1, 2, ..., N\}$ of cardinality K, i.e, $|\Lambda| = K$,*

$$\|\eta_\Lambda\|_{\ell_1} \leq \gamma \|\eta_{\Lambda^C}\|_{\ell_1}$$

Note that, by definition this implies that there is no $2K$ sparse vector that exists in the null-space of $\mathbf{\Phi}$. Therefore, the null-space property of order K implies that $Spark(\mathbf{\Phi}) > 2K$. It also implies that the sorted coefficients of the null-space vectors cannot decay rapidly to zeros.

3.2.3 Restricted Isometry Property (RIP)

In order to reconstruct the signal $\mathbf{x}^*$ in Eqn. (3.6) with high accuracy, the measurement matrix Φ has to satisfy the *Restricted Isometry Property* (RIP) [17] which is defined as follows.

Definition of RIP. *The measurement matrix Φ is said to satisfy RIP of order K with constant δ_K if and only if $\forall \|\boldsymbol{x}\|_{\ell_0} \leq K$, we have the following condition:*

$$(1 - \delta_K)\|\boldsymbol{x}\|_{\ell_2}^2 \leq \|\Phi\boldsymbol{x}\|_{\ell_2}^2 \leq (1 + \delta_K)\|\boldsymbol{x}\|_{\ell_2}^2.$$

In this definition, the lengths of all K-sparse vectors are approximately preserved. The definition becomes meaningful when it satisfies three following conditions. Firstly, δ_K has to be smaller than one, i.e. $\delta_K < 1$. Secondly, $\delta_K < 1$ implies that no vector in the null-space $N(\Phi)$ is K-sparse. Therefore, if we have $\delta_K < 1$, then we can conclude that $Spark(\Phi) > K$. Finally, if $\delta_K < 1$, then Φ is a one-to-one map on all $K/2$-sparse vectors. That is, no two $K/2$-sparse vector map to same point under action of Φ.

Lemma of RIP. *Let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ have disjoint supports. If $\|\boldsymbol{x}_1\|_{\ell_0} = K_1$ and $\|\boldsymbol{x}_2\|_{\ell_0} = K_2$, and Φ satisfies RIP of order $K_1 + K_2$, then $\|\Phi\boldsymbol{x}_1, \Phi\boldsymbol{x}_2\| \leq \theta_{K_1+K_2}\|\boldsymbol{x}_1\|\|\boldsymbol{x}_2\|$.*

Proof. The proof of this Lemma can be shown as follows:
Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be unit-norm. Noting that $\mathbf{x}_1 \pm \mathbf{x}_2$ are $(K_1 + K_2)$-sparse. Using the RIP definition, we have:

$$2(1 - \delta_{K_1+K_2}) \leq \|\Phi(\mathbf{x}_1 \pm \mathbf{x}_2)\|^2 \leq 2(1 + \delta_{K_1+K_2})$$

Now,

$$\langle \Phi\mathbf{x}_1, \Phi\mathbf{x}_2 \rangle = (\|\Phi(\mathbf{x}_1 + \mathbf{x}_2)\|^2 - \|\Phi(\mathbf{x}_1 - \mathbf{x}_2)\|^2)/4$$

Hence,

$$\langle \Phi \mathbf{x}_1, \Phi \mathbf{x}_2 \rangle \leq (2(1 + \delta_{K_1+K_2}) - 2(1 - \delta_{K_1+K_2}))/4 = \delta_{K_1+K_2}$$

Similarly,

$$\langle \Phi \mathbf{x}_1, \Phi \mathbf{x}_2 \rangle \geq (2(1 - \delta_{K_1+K_2}) - 2(1 + \delta_{K_1+K_2}))/4 = -\delta_{K_1+K_2}$$

□

3.2.4 From RIP to NSP

The constraint in the RIP condition is more powerful than the one in NSP. Given a matrix Φ, if it satisfies the RIP condition, then it also satisfies the NSP condition. The relationship between RIP and NSP is concisely shown by the following theorem.

Theorem (Candes, 2008) 1. *If the measurement matrix Φ satisfies RIP of order $2K$, then it also satisfies NSP of order K with*

$$\gamma = \frac{\sqrt{2}\delta_{2K}}{1 - \delta_{2K}}$$

Proof. Let $\eta \in N(\Phi)$. Let T_0 be the support of largest K entries of η, T_1 the next K largest entries, T_2 the next set, and so on. Therefore, T_j are all disjoint.

$$(1 - \delta_{2K}) \|\eta_{T_0 \cup T_1}\|^2 \leq \|\Phi \eta_{T_0 \cup T_1}\|^2$$

$$\|\Phi \eta_{T_0 \cup T_1}\|^2 = \langle \Phi \eta_{T_0 \cup T_1}, -\Phi \eta_{(T_0 \cup T_1)^C} \rangle$$

$$= \langle \Phi \eta_{T_0} + \Phi \eta_{T_1}, -\sum_{j \geq 2} \Phi \eta_{T_j} \rangle$$

$$= \langle \Phi\eta_{T_0} + \Phi\eta_{T_1}, -\sum_{j\geq 2} \Phi\eta_{T_j} \rangle$$

$$\leq \delta_{2K}(\|\eta_{T_0}\| + \|\eta_{T_1}\|) \sum_{j\geq 2} \|\eta_{T_j}\|$$

$$\leq \sqrt{2}\delta_{2K}\|\eta_{T_0\cup T_1}\| \sum_{j\geq 2} \|\eta_{T_j}\|$$

Note that each entry of η_{T_j} is less than or equal to any entry of $\eta_{T_{j-1}}$. Therefore, each entry of η_{T_j} is less than $\|\eta_{T_{j-1}}\|_{\ell_1}/K$. Therefore, $\|\eta_{T_j}\| \leq \|\eta_{T_{j-1}}\|_{\ell_1}/\sqrt{K}$.

Since η_{T_0} and η_{T_1} are orthogonal, $\|\eta_{T_0}\| + \|\eta_{T_1}\| \leq \sqrt{2}\|\eta_{T_0\cup T_1}\|$. Hence, we obtain:

$$\|\Phi\eta_{T_0\cup T_1}\|^2 \leq \sqrt{2}\delta_{2K}\|\eta_{T_0\cup T_1}\| \sum_{j\geq 2} \|\eta_{T_{j-1}}\|_1/\sqrt{K}$$

$$\leq \sqrt{2}\delta_{2K}\|\eta_{T_0\cup T_1}\|\|\eta_{T_0^C}\|_1/\sqrt{K}$$

Putting it all together,

$$\|\eta_{T_0}\|_{\ell_1} \leq \sqrt{K}\|\eta_{T_0}\|_2 \leq \sqrt{K}\|\eta_{T_0\cup T_1}\| \leq \frac{\sqrt{2}\delta_{2K}}{1-\delta_{2K}}\|\eta_{T_0^C}\|_{\ell_1}$$

□

3.2.5 Random Projections for Compression

Given N data points with high dimensions in an Euclidian space $\mathbb{R}^d$, the random projection method allows embedding these points *randomly* in some low dimension, logarithmic in N and *independent* of d, without distorting the pairwise squared distances between the points by more than a factor of $1 \pm \epsilon$ (as shown in Figure 3.1). Unlike other regular dimensionality reduction methods that usually depend on the number of training samples and their dimensionality, i.e. PCA [55, 105], LDA [8], LPP [51], etc., this method is still able to reduce the dimensions of training data without having to worry about the number of samples available. It has been considered as an efficient dimensionality reduction method

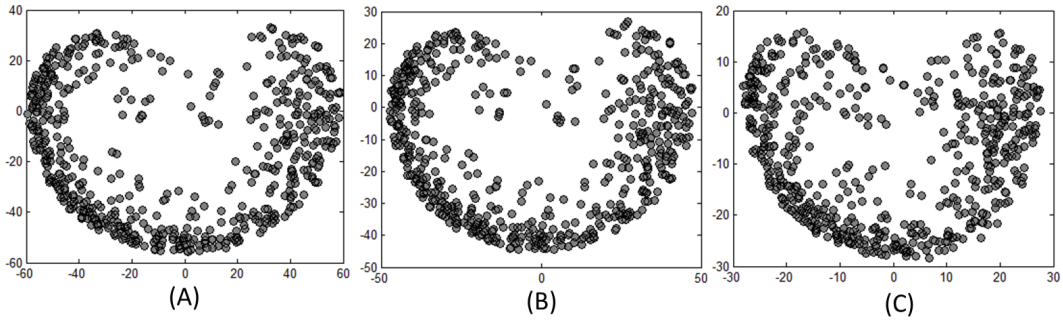


Figure 3.1: Examples of RP for dimensionality reduction. (A) the first two eigenvectors trained from all images of the first subject in the Extended Yale-B database, (B) the first two eigenvectors trained from the images in (A) projected on RP subspace at 50% of the original energy, (C) the first two eigenvectors trained from those images projected on RP subspace at only 10% of the original energy.

in many papers [12, 54]. The idea of the random projection is detailed in the following Johnson-Linderstrauss lemma [54].

Lemma (Johnson-Lindenstrauss) [54]. *Given a set* $\boldsymbol{X}$ *of* N *points* $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \in \mathbb{R}^d$ *and any scalar* ϵ*, where* $\epsilon \in (0, \frac{1}{2})$*,* $\boldsymbol{X}$ *is projected onto an uniform random* k*-dimensional subspace where*

$$k \geq \frac{9 \ln N}{\epsilon^2 - \frac{2}{3}\epsilon^3} + 1 = O(\epsilon^{-2} \ln N)$$

$\forall \boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{X}$*, the following property is achieved with probability at least* $\frac{1}{2}$*:*

$$(1 - \epsilon)\|\boldsymbol{u} - \boldsymbol{v}\|^2 \leq \|\phi(\boldsymbol{u}) - \phi(\boldsymbol{v})\|^2 \leq (1 + \epsilon)\|\boldsymbol{u} - \boldsymbol{v}\|^2$$

where $\phi(\mathbf{u})$, $\phi(\mathbf{v})$ are the projections of $\mathbf{u}$, $\mathbf{v}$, Lipschitz mapping from a high dimensional space $\mathbb{R}^d$ into a low dimensional one $\mathbb{R}^k$. The proof of this lemma can be found in [26]. More precisely, this lemma is deduced from another lemma on norm preservation, described as follows:

Lemma (Norm preservation [35]). *Let* $\boldsymbol{x} \in \mathbb{R}^d$*, assume that all the entries of the random matrix* $\Phi \in \mathbb{R}^{k \times d}$ *are sampled independent and identically distributed from a standard*

normal distribution $N(0, 1/k)$*, then the following property holds:*

$$Pr((1-\epsilon)\|\boldsymbol{x}\|^2 \leq \|\frac{1}{\sqrt{k}}\Phi\boldsymbol{x}\|^2 \leq (1+\epsilon)\|\boldsymbol{x}\|^2) \geq 1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}$$

From this lemma, it can be concluded that there is only a need of $k = C/(\epsilon^2 - \epsilon^3)$ in order to guarantee the existence of a measurement matrix Φ nearly isometric on **x** with a positive probability. When C large enough, there is a very small probability that the norm preservation condition is violated. In practice, most of random Gaussian or Bernoulli matrices with a considerable number of rows all satisfy the condition of norm preservation. In addition, it has been observed that for random matrices with entries of random variables following the i.i.d sub-Gaussian distribution or any zero-mean random variable with a bounded distribution, a random variable a satisfies $Pr(|a| \geq T) = 0$.

There is a strong relationship between the RIP condition and the Johnson-Lindenstrauss lemma discussed above. Most random projection methods used to map data points from a high dimensional space into a linear and distance preserving space are also able to generate a matrix that satisfies the RIP condition. The following theorem shows the conditions necessary for random matrices to satisfy RIP.

Theorem (Baraniuk et al., 2008). *Let the entries of the random matrix* $\Phi \in \mathbb{R}^{k\times d}$ *be independent and identically distributed sampled from a standard normal distribution* $N(0, 1/k)$*. If*

$$k = Cs \log \frac{d}{s}$$

then the random matrix Φ *is near-isometric with constant* ϵ *on all* s*-sparse vectors with a probability* $\geq 1 - e^{-C_2 k}$*.*

This theorem can also be applied on s-sparse signals obtained from a transform basis, e.g. wavelet, Hadamard, etc. In addition, the RIP conditions can be violated with a probability that exponentially reduces in s.

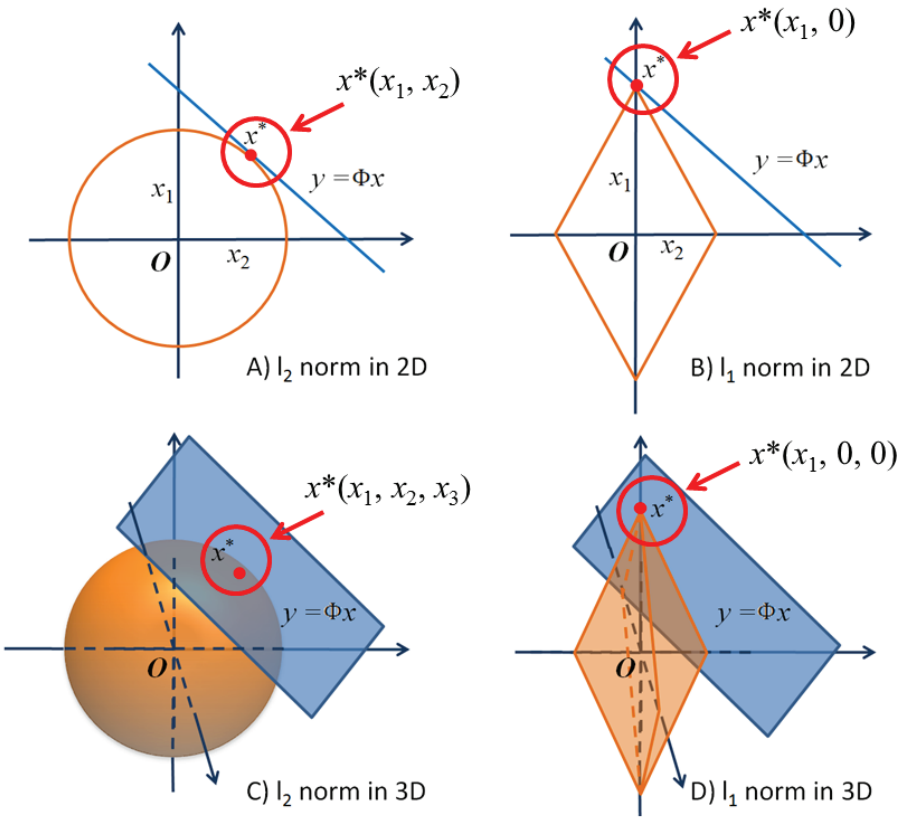


Figure 3.2: Illustration of the solution x^* in cases: A) ℓ_2-norm in 2D, B) ℓ_1-norm in 2D, C) ℓ_2-norm in 3D and D) ℓ_1-norm in 3D.

3.3 ℓ_p-norm Minimization

In this section, we review the intuition behind using the ℓ_1-norm in compressed sensing problems. Generally, the ℓ_p-norm minimization problem defined in Eqn. (3.6) can be defined as follows:

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_p} \qquad \text{subject to} \quad \mathbf{y} = \Phi\mathbf{x} \tag{3.12}$$

Depending on an assigned value of p, we can find the solution in Eqn. (3.12) in different ways. When $p < 1$, the equation is no longer a formal norm. Therefore, these cases will not be considered in our problems. In this section, we will analyze this equation when p is equal to zero, one, and two.

3.3.1 ℓ_2-norm Minimization

When the value of p is assigned to two, $p = 2$, the Eqn. (3.12) becomes the Least Square Error (LSE) or ℓ_2-norm $\|\mathbf{x}\|_{\ell_2}$, or $\|\mathbf{x}\|$ and is defined as follows:

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_2} \qquad \text{subject to} \quad \mathbf{y} = \Phi\mathbf{x} \tag{3.13}$$

or

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_2}^2 \qquad \text{subject to} \quad \mathbf{y} = \Phi\mathbf{x} \tag{3.14}$$

The method of Lagrange multipliers can be applied to solve this problem. It is defined as follows: [35]:

$$\mathbf{\Gamma}(\mathbf{x}) \triangleq \|\mathbf{x}\|_{\ell_2}^2 + \boldsymbol{\lambda}^T(\Phi\mathbf{x} - \mathbf{y}) \tag{3.15}$$

where $\boldsymbol{\lambda}$ is the Lagrange multipliers for the constraint set. Then, we take the derivative of $\mathbf{\Gamma}(\mathbf{x})$ with respect to $\mathbf{x}$ and set it to zero in order to find the optimal point:

$$\frac{\partial \mathbf{\Gamma}(\mathbf{x})}{\partial \mathbf{x}} = 2\mathbf{x} + \boldsymbol{\lambda}^T\Phi = 0 \tag{3.16}$$

Then, the solution $\mathbf{x}^*$ is computed as follows:

$$\mathbf{x}^* = -\frac{1}{2}\mathbf{\Phi}^T\boldsymbol{\lambda} \tag{3.17}$$

from Eqn. (3.14) and (3.17), we have:

$$\mathbf{y} = \Phi\mathbf{x}^* = -\frac{1}{2}\Phi\Phi^T\boldsymbol{\lambda} \tag{3.18}$$

Then, $\boldsymbol{\lambda}$ can be found as follows:

$$\boldsymbol{\lambda} = -2(\Phi\Phi^T)^{-1}\mathbf{y} \tag{3.19}$$

Finally, the solution $\mathbf{x}^*$ can be found:

$$\mathbf{x}^* = -\frac{1}{2}\Phi^T\boldsymbol{\lambda} = \Phi^T(\Phi\Phi^T)^{-1}\mathbf{y} \tag{3.20}$$

This is the closed-form pseudo-inverse solution. Φ is known as a full-rank matrix for solving a strictly convex problem. Using this form, we can easily find a unique solution.

3.3.2 ℓ_1-norm Minimization

When the value of p is assigned to one, $p = 1$, the Eqn. (3.12) becomes the ℓ_1-norm problem that we are interested in:

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_1} \qquad \text{subject to} \quad \mathbf{y} = \Phi\mathbf{x}. \tag{3.21}$$

where the ℓ_1-norm is defined as:

$$\|\mathbf{x}\|_{\ell_1} \triangleq \sum_i |\mathbf{x}_i| \tag{3.22}$$

Eqn. (3.21) is clearly convex but not strictly so. Additionally, it may have more than one solution. It can be claimed that these solutions are gathered in a set that is bounded and convex, and there always exists at least one with at most N non-zeros among these solutions.

As discussed above, it can be claimed that ℓ_1-norm always gives a sparser solution than ℓ_2-norm does. This idea is illustrated in Figure 3.2. Imagine that the constraint on the subspace $\mathbf{y} = \Phi\mathbf{x}$ is a two dimensional (2D) line in 2D cases or a plane in 3D cases. In Figures 3.2.A and 3.2.C, the solution $\mathbf{x}^*$ of ℓ_2 norm is the intersection point between the line (or the plane) and the circle (or the ball). In this case, the solution points $\mathbf{x}^*(x_1, x_2)$ (or $\mathbf{x}^*(x_1, x_2, , x_3)$), where $x_i \neq 0$, are mostly valid. In other words, the solutions of ℓ_2-norm are not frequently sparse. However, for the case ℓ_1-norm (Figures 3.2.B and 3.2.D), the

solution points (or the intersection) $\mathbf{x}^*$ usually stay on the axes. Therefore, there is at least one value in 2D cases (or two values in 3D cases) with $x_i = 0$. This provides some intuition for why the solutions of ℓ_1-norm are always sparser than the ones from ℓ_2-norm.

There are numerous algorithms that pursue ℓ_1 minimization, i.e. Orthogonal Matching Pursuit (OMP), Gradient Projection [42, 83], Homotopy [30, 76], Iterative Thresholding [116], Augmented Lagrangian [122], etc. One of the most fundamental and popular algorithms is the Orthogonal Matching Pursuit which is presented in detail in [35].

3.3.3 ℓ_0-norm Minimization

When the value of p is assigned to zero, $p = 0$, the Eqn. (3.12) becomes the ℓ_0-norm problem and defined as follows:

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}} \|\mathbf{x}\|_{\ell_0} \qquad \text{subject to} \quad \mathbf{y} = \Phi\mathbf{x} \tag{3.23}$$

where $\|\mathbf{x}\|_{\ell_0}$ is defined as follows:

$$\|\mathbf{x}\|_{\ell_0} \triangleq \lim_{p\to 0} \|\mathbf{x}\|_{\ell_p}^p = \lim_{p\to 0} \sum_{k=1}^{m} |x_k|^p = \#(i : x_i \neq 0) \tag{3.24}$$

In this case, the sparsity of a given vector $\mathbf{x}$ can be simply calculated by *counting* the number of non-zeros entries intuitively. It can be seen that the ℓ_0-norm provides a very simple and easily grasped notion of sparsity. However, it is not a practical solution, especially when we need to solve a vector $\mathbf{x}$ in a very huge dimensional space.

Beyond conceptual issues of uniqueness and verification of solutions, one is easily overwhelmed by the apparent difficulty of solving this problem. This is a classical problem of combinatorial search. One sweeps exhaustively through all possible sparse subsets, generating corresponding subsystems $\mathbf{y} = \Phi_S\mathbf{x}_S$ where Φ_S denotes the matrix having $|S|$ columns chosen from those columns of Φ with indices in S and checking if $\mathbf{y} = \Phi_S\mathbf{x}_S$ can be solved. The complexity of the exhausted search is exponential in M, and indeed, it has

been proven that ℓ_0 is, in general, NP-Hard. Tao et al. [101] have proved that this ℓ_0-norm problem can be replaced by the norm problem of ℓ_1 which can be easy to find solution.

Chapter 4

Compressed Submanifold Multilinear Analysis (CSMA)

Multifactor analysis plays an important role in data analysis since most real-world datasets usually exist under the combination of numerous factors which are usually not independent but interdependent together. Multilinear Principal Component Analysis, or Tensorfaces [108, 110] has been become one of the leading multilinear analysis methods since last decade. The heart of Multilinear PCA is to use Principal Component Analysis and Higher-order SVD to decompose factors from a given tensor [58, 62]. However, since this method is developed from the regular multilinear algebra and the ℓ_2-norm Higher-order SVD to construct the relationships among factors of given data, it is therefore impossible to deal with missing values in the data. In addition, it is also very sensitive against noises usually existed in practice. Finally, this method also has very expensive computation when dealing with huge multi-dimensional data for dimensionality reduction and averaging factors among submanifolds. In order to overcome these limitations, we propose a novel method named Compressed Submanifold Multilinear Analysis (CSMA) that allows handling missing values. Given a tensor with missing values, our method is still able to decompose its factors and presents them in a meaningful way. Our proposed method also has ability to

detect outliers and decompose tensors robustly against noises. The method keeps the full use of individuals submanifold without running any averaging process. In the first section of this chapter, we review the limitations as well as the motivation for the current multi-factor decomposition methods. Then, we present our proposed CSMA method in details in the second section of the chapter.

4.1 Motivation of CSMA

In this section, we first review and demonstrate the disadvantages in the classical Multilinear PCA method. We then show the motivation to develop our novel method named Compressed Submanifold Multilinear Analysis (CSMA) in order to overcome these limitations.

4.1.1 Limitations of Multilinear PCA

Although widely used, Multilinear PCA, one of the leading multilinear analysis methods, still suffers from three major drawbacks. First, it is very sensitive to outliers and noise and unable to cope with missing values. Second, since MPCA deals with huge multi-dimensional datasets, it is usually computationally expensive. Finally, it loses original local geometry structures due to the averaging process.

Missing Values and Outliers

Since the heart of the Multilinear PCA is employed by using Principal Component Analysis and Higher-order SVD to decompose factors from a given tensor, the efficiency of the decomposition is therefore much depend on the SVD process. However, the regular SVD method is very sensitive to outliers and noisy values usually existed in given data due to the equal treatment for all input data points and the disability in detecting outliers of the regular ℓ_2-norm computation. The problem of the SVD subspace estimation defined in

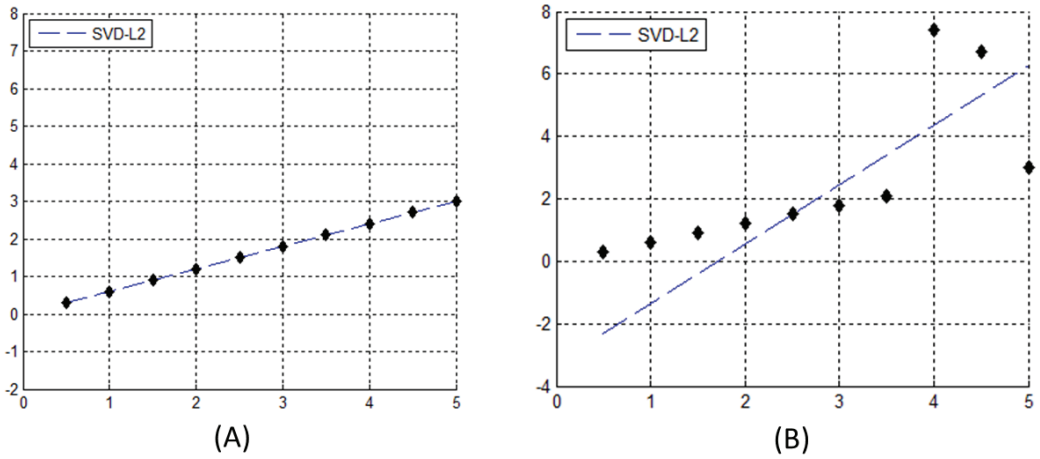


Figure 4.1: An example to show the limitation of the classical SVD method. (A) The classical SVD method can present good enough the subspace when the input data doesn't contain any noisy values or outliers. (B) However, when the data have some outliers, the represented subspace will be affected. It happens because the classical SVD is very sensitive to outliers and noises.

section 2.4.1 becomes equivalent to the optimization of the cost function ε that can be solved by using ℓ_2-norm as follows:

$$\begin{aligned}\varepsilon(\mathbf{U},\mathbf{V}) &= \|\mathbf{X}_{d\times n} - \mathbf{U}_{d\times r}\mathbf{\Sigma}_{r\times r}\mathbf{V}^{\top}_{n\times r})\|^2_{\ell_2} \\ &= \sum_{i=1}^{d}\sum_{j=1}^{n}(\mathbf{x}_{i,j} - \mathbf{u}_i\mathbf{v}_j^{\top})^2\end{aligned}$$

where the matrix $\mathbf{X}_{d\times n}$ is defined as in Eqn. (2.19). The presented equation describes the first r columns of the orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$ representing a subspace to minimize the defined ℓ_2-norm cost function. This truncated process can be denoted as $\mathbf{U} = \mathbf{U}(:,1:r)$ and $\mathbf{V} = \mathbf{V}(:,1:r)$. The SVD solution can be simply solved in a regular closed form with the given ℓ_2-norm cost function. As shown in section 3.3, the ℓ_2-norm process however treats all input data equally and doesn't have ability to detect the outliers (or sparse components). Therefore, Multilinear PCA is usually sensitive to outliers and noisy values from given input data. Figure 4.1 shows an example of the limitations in SVD representation. When input data is good enough (without noises or outliers), the SVD

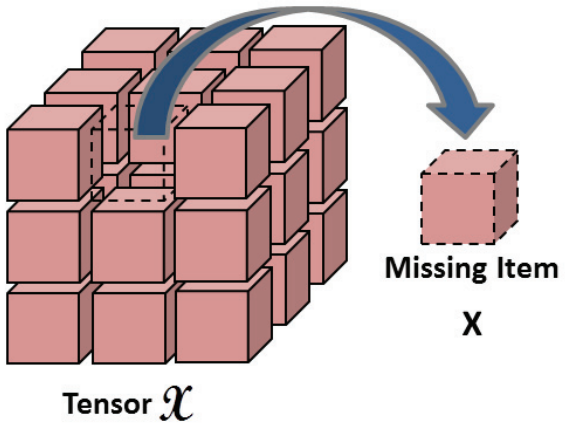


Figure 4.2: An example of three-order tensor $\mathcal{X} \in \mathbb{R}^{3\times3\times3}$ with a missing value $\mathbf{X}$. Notice that $\mathbf{X}$ may only miss some of its dimensions, i.e. $\mathbf{X}_{:,j,k}$ or $\mathbf{X}_{:,:,k}$ or all of its dimensions $\mathbf{X}_{i,j,k}$.

method can generate a good subspace to represent the data distribution. However, when the data contains some noises or outliers, this subspace easily receives a structure distortion due to these noises. Therefore it doesn't present well enough the data distribution.

In addition, as presented in Eqn. (2.19), there is no mechanism to denote missing values in the regular SVD representation. The input matrix $\mathbf{X}$ that need to be decomposed must be filled with values for all $d \times n$ items. Otherwise, it becomes unsolvable. Because of this, the Multilinear PCA method is clearly unable to decompose any given tensor accompanied with missing values as shown in Figure 4.2.

Tensors with High Dimensions

The complexity of Multilinear PCA can be counted on the computational time used in matrix multiplication and Singular Value Decomposition computation steps. In general, it requires to spend $O(m \times n \times d)$ time on multiplication of two matrices $\mathbf{X}$ of sizes $m \times n$ and $\mathbf{Y}$ of size $n \times d$. Meanwhile, given a matrix $\mathbf{X}$ of sizes $m \times n$, it takes $O(\max(m,n) \times \min(m,n)^2)$ time to compute its SVD formulation.

Since SVD consumes much more time than matrix multiplication, we consider the time the method spends only for SVD, not for matrix multiplication, during the training step.

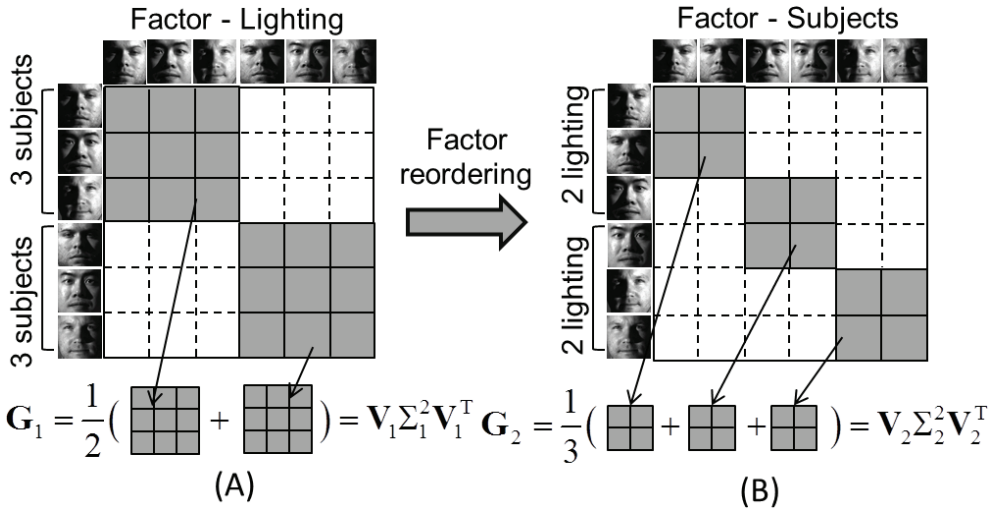


Figure 4.3: The illustration to averaging process with 6 training images of 3 subjects and 2 lighting conditions. The 6×6 matrix is the Gram matrix of the *reordered* images with an appropriate permutation matrix. In (A), two 3×3 blocks in grey are the Gram matrices of 2 lightings. Each of two subsets consists of 3 subjects' faces under each of 2 lightings. The averaging of the Gram matrix $\mathbf{G}_1$ of these 3×3 block matrices in grey presents the average dot products among 3 subjects across 2 lightings. This process is applied similarly for the subject factor in (B).

First of all, PCA performs the SVM operation a single time when this method computes the SVD of $X \in R^{d \times n}$ to compute the linear transformation matrix U. It often occurs that d is greater than n and in this case, PCA requires $O(d \times n^2)$ to learn a training set. The matrix U can be calculated using $n \times n$ Gram matrix $X^T X$ as well as using X, and PCA's training time can be reduced to $O(n^3)$.

Calculating U is necessary in MPCA. Additionally, MPCA uses SVD to calculate V_i from $X^{(k)}X^{(k)T} \in R^{n_k \times n_k}$ in the three-factor framework. Thus MPCA runs in $T(n_1^3) + T(n_2^3) + T(n_3^3)$ more time than it takes PCA to run. In big O notation, MPCA takes $O(n^3)$ to run.

Local Geometry Structure Averaging

One of the most important properties in Principal Component Analysis is that the low-dimensional subspaces obtained by this method always preserve the dot products between pairs of samples. As shown in Eqn. (4.1), given a matrix $\mathbf{X}$, the Gram matrix of $\mathbf{X}$ and

the Gram matrix of the PCA projected components $\mathbf{Y}_{PCA} = \mathbf{U}^\top\mathbf{X} = \mathbf{\Sigma}\mathbf{V}^\top$ are always identical. This property can be illustrated as in Eqn. (4.1).

$$\begin{aligned}
\mathbf{G}_\mathbf{X} &= \mathbf{X}^\top\mathbf{X} = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top)^\top(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top) \\
&= \mathbf{V}\mathbf{\Sigma}\mathbf{U}^\top\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top \\
\mathbf{G}_{\mathbf{Y}_{PCA}} &= \mathbf{Y}_{PCA}^\top\mathbf{Y}_{PCA} = (\mathbf{\Sigma}\mathbf{V}^\top)^\top(\mathbf{\Sigma}\mathbf{V}^\top) \\
&= \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^\top
\end{aligned} \tag{4.1}$$

In order to further understand the properties of the decomposed parameters in Multilinear PCA, another equivalent approach can be shown to compute these parameters. In this approach, the process of the Multilinear PCA computation is employed based on averaging parts of the Gram matrix as illustrated in Fig. 4.3 [80]. When $i = 1$, the Gram matrix $\mathbf{G}_1$ of the first factor can be computed as,

$$\mathbf{G}_1 = \frac{1}{n_2 n_3}\sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3} G(\mathbf{X}(:, i_2, i_3)) \tag{4.2}$$

$\mathbf{G}_1$ is the mean of all $\frac{1}{n_1}\prod_{i=1}^{N} n_i$ Gram matrices of factor-1-variant subsets and $\mathbf{X}_1 = \mathbf{X}(:, i_2, i_3)$. A similar approach can be employed to find the other Gram matrices $\mathbf{G}_i$. The Gram matrix denotes the Euclidean distances of all sample pairs. Therefore, factor-i parameters computed by $\mathbf{G}_i$ show the mean structure of the $\frac{1}{n_i}\prod_{i=1}^{N} n_i$ factor-i-variant submanifolds. MPCA aims to independently decompose the relationships of multifactors through the averaging process. From Eqns. 4.1 and 4.2, the Gram matrix $\mathbf{G}_1$ can be simply derived as follows,

$$\begin{aligned}
\mathbf{G}_1 &= \frac{1}{n_2 n_3}\sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3} G(\mathbf{X}(:, i_2, i_3)) \\
&= \mathbf{V}_1\mathbf{\Sigma}_1^2\mathbf{V}_1^\top
\end{aligned} \tag{4.3}$$

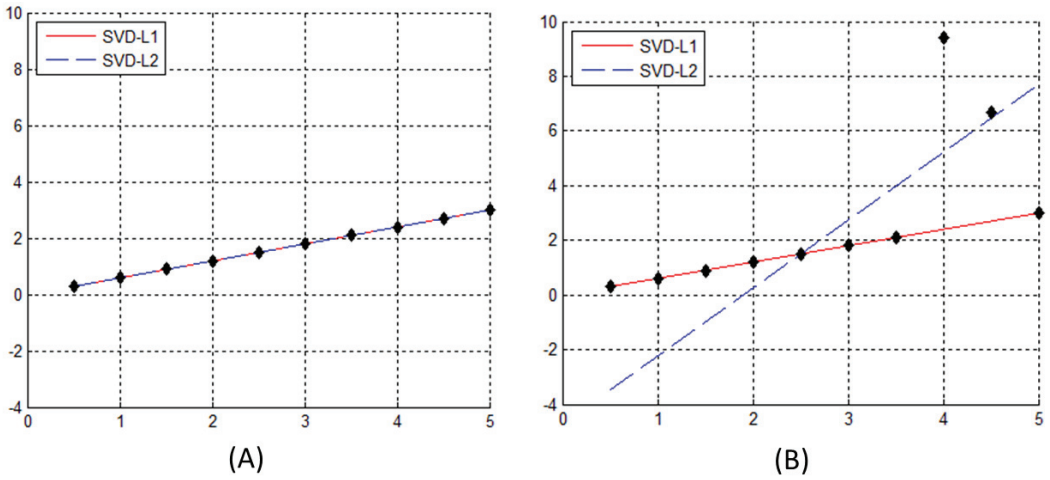


Figure 4.4: A comparison between the classical SVD method and the SVD-ℓ_1 method. (A) Both methods fit very well on the subspace when the input data doesn't contain any noisy values or outliers. (B) When the data contains some outliers, the SVD subspace will be affected, meanwhile the SVD-ℓ_1 subspace still represents good enough the subspace. It happens because the SVD-ℓ_1 is robust against noises and outliers.

Therefore, in order to avoid the averaging process when computing Gram matrices $\mathbf{G}_i$, [80] uses the columns of $\mathbf{\Sigma_i V}_i^\top$ to compute it. Procrustes is then employed for local alignment to remove the differences in each coordinate while preserving structures of each manifold. However, although exploring the submanifold property, this method still remains with two major drawbacks. First, this method is unable to deal with missing values. In addition, it also doesn't allow the high accuracy in local alignment. Therefore, our solution to remedy these drawbacks will be introduced in the following sections.

4.1.2 Innovations in CSMA

The limitations presented in section 4.1.1 have become the motivation for us to develop a new tensor approach to overcome these disadvantages. In this section, we outline the advantages in our proposed method compared to the standard deterministic Multilinear PCA algorithm. The details of our algorithms can be found in next few sections.

Tensor Completion with ℓ_1-norm Solutions

The regular ℓ_2-norm solution still remains three limitations as discussed in section 4.1.1. Instead, we propose to solve the Higher-order SVD in a newly defined framework with the support of ℓ_1-norm optimization. Indeed, section 3.3 shows that ℓ_1-norm representation is able to efficiently detect outliers and clearly robust against noises. Figure 4.4 presents an example to show the advantages of our proposed SVD-ℓ_1 subspace representation compared to the one derived from ℓ_2-norm. When there are no outliers or noises existed in a given data, the SVD-ℓ_1 method can generate a subspace as good as the one created by the SVD-ℓ_2. However, when the given data contains some outliers or noises, the subspace generated by the SVD-ℓ_1 method is more robust than the one created by the SVD-ℓ_2.

In addition, far apart from the traditional Multilinear PCA, our proposed framework allows presenting input tensors with missing values efficiently. Since our method has ability to present missing values and complete missing holes in given tensors, it therefore can be named as a tensor completion method. Our proposed framework is then solved using the Alternative Direction Method of Multipliers (ADMM) method [13, 124] that can be used efficiently to solve the multi-variable optimization problems.

Low-Rank Approximation with Random Projection

Derived from Johnson-Linderstrauss lemma, Random Projection can be clearly employed in order to speed up the computation in multifactor Singular Value Decomposition with both ℓ_1-norm and ℓ_2-norm cases. In worst cases, the standard ℓ_2-norm based Singular Value Decomposition low-rank approximation can take up to $O(dn^2)$ time to decompose a matrix, where ($d > n$). Meanwhile, Random Projection can help to speed it up to $O(dn \log n)$ time. However, in practical applications, input matrices are mostly k-sparse. In other words, these given matrices usually have at least m non-zeros entries. In that case, standard Singular Value Decomposition methods may consume $O(mn)$ time while Random Projection takes only $O(m \log n + n \log^2 n)$ time [111].

Local Coordinate Alignment

Park and Savvides [78, 80] presented a procrustes analysis [45] approach to avoid the averaging distortion when decomposing a tensor. However, this method is unusable when solving the tensors with missing values because its formula doesn't allow representing missing values. We therefore, present a Stratified method [7, 84] that can help to avoid the averaging process to align the local coordinates while still able to preserve submanifold structures. More importantly, compared to other alignment methods, this approach is able to deal with missing and noisy data. The proposed local coordinate alignment method will be presented in detail in Section 4.4 of this chapter.

4.2 Multifactor ℓ_1-based Decomposition

There are numerous methods that have ability to deal with missing data and outliers in the problem of 2D matrix decomposition, to cite a few [39, 57, 61, 64, 99, 128]. However, in this section, we are going to present a new SVD-ℓ_1 formula that can be incorporated into the multifactor framework to decompose factors. This method can converge very fast when pursuing the Alternative Direction Method of Multipliers. In addition, this SVD-ℓ_1 algorithm is also able to decompose relationships among factors from a given tensor. In this section, we first introduce the proposed SVD-ℓ_1 formula. Then, we present how to use the Alternative Direction Method of Multipliers to solve the problem. Finally, the ℓ_1-norm based Higher-order SVD (HOSVD-ℓ_1) is also presented. We demonstrate how to use the HOSVD-ℓ_1 to handle the multifactor data to cope with missing values.

4.2.1 SVD-ℓ_1 Reformulation

Given a matrix $\mathbf{X}_k \in \mathbb{R}^{d \times n}$ which is flattened from a tensor $\mathcal{X}$ along the factor k, instead of using the regular SVD method as defined in section 2.4.1, it can be reformulated by solving the ℓ_1-norm problem in addition to the defined weight parameters. This SVD-ℓ_1 problem

can be solved by minimizing the ℓ_1-norm error with the constraints as follows,

$$\begin{aligned} \min_{\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k} \quad & \|\mathbf{W}_k \odot (\mathbf{X}_k - \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top)\|_{\ell_1} \\ s.t., \quad & \mathbf{U}_k^\top \mathbf{U}_k = \mathbf{I}, \mathbf{V}_k^\top \mathbf{V}_k = \mathbf{I} \end{aligned} \tag{4.4}$$

or

$$\begin{aligned} \min_{\mathbf{u}_i^{(k)}, \lambda_i^{(k)}, \mathbf{v}_i^{(k)}} \quad & \sum_{i=1}^{r} \|\mathbf{w}_i^{(k)} \odot (\mathbf{X}_k - \lambda_i^{(k)} \mathbf{u}_i^{(k)} \mathbf{v}_i^{(k)\top})\|_{\ell_1} \\ s.t., \quad & \mathbf{u}_i^{(k)\top} \mathbf{u}_i^{(k)} = \mathbf{I}, \mathbf{v}_i^{(k)\top} \mathbf{v}_i^{(k)} = \mathbf{I} \end{aligned} \tag{4.5}$$

where all symbols are defined similarly to the ones in Eqn. (2.19). Far apart from SVD-ℓ_2, our defined SVD-ℓ_1 formula allows to decompose $\mathbf{X}_k$ with missing values and outliers denoted by the weight matrix $\mathbf{W}_k$ flattened along the factor k from the weight tensor $\mathcal{W}$. In the given matrix $\mathbf{X}_k$, if the value of the data point at the position (i, j) exists, its weight is set to a positive number, $\mathbf{W}_k(i, j) > 0$. Otherwise, if the value of the data point at the position (i, j) is missing, its weight is set to zero, $\mathbf{W}_k(i, j) = 0$. The error between the matrix $\mathbf{X}_k$ and reconstruction from the SVD-ℓ_1 is minimized by using the ℓ_1-norm distance. It is the key idea that allows the reconstruction results in our method robust against noises and outliers.

In the Eqn. (4.4), the $\odot$ symbol denotes the component-wise multiplication. $\mathbf{U}_k$ and $\mathbf{V}_k$ are two decomposed matrices. The constraints $\mathbf{U}_k^\top \mathbf{U}_k = \mathbf{I}$ and $\mathbf{V}_k^\top \mathbf{V}_k = \mathbf{I}$ are to guarantee both two matrices $\mathbf{U}_k$ and $\mathbf{V}_k$ are all orthonormal, where each has the length of 1 and every pair is orthogonal, as the ones in the standard SVD method. $\mathbf{\Sigma}_k$ is a diagonal matrix containing the square root of eigenvalues from $\mathbf{U}_k$ or $\mathbf{V}_k$ in descending order. r denotes the rank of $\mathbf{X}_k$, where $r \leq \min(d, n)$, $\mathbf{u}_i^{(k)} \in \mathbb{R}^d$ and $\mathbf{v}_i^{(k)} \in \mathbb{R}^n$ also have the length of 1. Each pair of ($\mathbf{u}_i^{(k)}$, $\mathbf{v}_i^{(k)}$) denotes left and right singular vectors with singular value $\lambda_i^{(k)} > 0$, where $\lambda_j \geq \lambda_{j+1}, \forall 1 \leq k \leq r - 1$. It follows that each $\mathbf{u}_i^{(k)}$ is an eigenvector of $\mathbf{X}_k \mathbf{X}_k^\top$

and $\mathbf{v}_i^{(k)}$ is an eigenvector of $\mathbf{X}_k^\top \mathbf{X}_k$, and the corresponding eigenvalues are $\lambda_i^{(k)2}$. In the following analysis, $\mathbf{X}$ is used in place of $\mathbf{X}_k$ for simplicity of notation.

4.2.2 Alternative Direction Method of Multipliers Solutions

The Eqn. (4.4) is unsolvable in general cases because it belongs to general non-convex problems. In that equation, we have to optimize three variables, i.e. $\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k$ and two orthogonal constraints. However, thanks to the recent advanced results from the optimization studies, especially the ALM/ADMM method, the minimization problem in Eqn. (4.4) can be solved in the following way. First, we redefine the problem in the form of trace norm regularization as shown in Eqn. (4.6),

$$\begin{aligned} \min_{\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k, \mathbf{E}} \quad & \|\mathbf{W} \odot (\mathbf{X} - \mathbf{E})\|_{\ell_1} + \lambda \|\mathbf{E}\|_* \\ s.t., \quad & \mathbf{E} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top, \\ & \mathbf{U}_k^\top \mathbf{U}_k = \mathbf{I}, \\ & \mathbf{V}_k^\top \mathbf{V}_k = \mathbf{I} \end{aligned} \tag{4.6}$$

The objective function defined in this minimization problem includes two main terms. The first ℓ_1-norm term aims to preserve the sparsity property in the reconstruction. Meanwhile the second trace norm term guarantees the low rank property of the solutions. The parameter λ controls the trade-off between trace norm regularization and reconstruction fidelity. In this equation, instead of optimizing three variables as in Eqn. (4.4), there are now four variables need to be optimized, i.e. $\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k, \mathbf{E}$. More importantly, let's analyze the property of $\mathbf{E}$. As defined in the constraints, we have $\mathbf{E} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top$. This equation can be further analyzed. In combining with two matrices $\mathbf{\Sigma}_k$ and $\mathbf{V}_k$, let $\tilde{\mathbf{U}}$ be a new orthogonal decomposed matrix computed from a matrix $\mathbf{V}$ by using Singular Value Decomposition method, i.e.,

$$[\tilde{\mathbf{U}}, \mathbf{\Sigma}_k, \mathbf{V}_k] = SVD(\mathbf{V}) \tag{4.7}$$

or

$$\mathbf{V} = \tilde{\mathbf{U}}\mathbf{\Sigma}_k\mathbf{V}_k \tag{4.8}$$

where $\tilde{\mathbf{U}}^\top\tilde{\mathbf{U}} = \mathbf{I}$. Given $\mathbf{U}_k$ and $\tilde{\mathbf{U}}$, it is easy to find a matrix $\mathbf{U}$ that qualifies the condition $\mathbf{U}_k = \mathbf{U}\tilde{\mathbf{U}}$, where $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. From these definitions, the matrix $\mathbf{E}$ can be redefined as follows:

$$\begin{aligned} \mathbf{E} &= \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top \\ &= \mathbf{U}\tilde{\mathbf{U}}\mathbf{\Sigma}_k\mathbf{V}_k^\top \\ &= \mathbf{U}\mathbf{V} \end{aligned} \tag{4.9}$$

It is noticed that the two matrices $\mathbf{U}$ and $\mathbf{V}$ in this section are different from those in section 4.1.1. In addition, since $\mathbf{U}$ is orthogonal, i.e. $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, its trace norm then becomes one, i.e. $\|\mathbf{U}\|_* = \|\mathbf{U}^\top\|_* = 1$. Therefore, the trace norm of $\mathbf{E}$ can be computed as follows,

$$\|\mathbf{E}\|_* = \|\mathbf{U}\mathbf{V}\|_* = \|\mathbf{V}\|_* \tag{4.10}$$

From the results shown in Eqns. (4.9) and (4.10), the problem (4.6) can be then simplified as in [128] as follows,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{E}} \quad & \|\mathbf{W} \odot (\mathbf{X} - \mathbf{E})\|_{\ell_1} + \lambda\|\mathbf{V}\|_* \\ s.t., \quad & \mathbf{E} = \mathbf{U}\mathbf{V}, \mathbf{U}^\top\mathbf{U} = \mathbf{I} \end{aligned} \tag{4.11}$$

There are now only three variables need to be optimized, i.e. $\mathbf{U}_k, \mathbf{V}_k, \mathbf{E}$ in this equation. More importantly, two defined terms in the objective function now have become independent, i.e. $\|\mathbf{W} \odot (\mathbf{X} - \mathbf{E})\|_1$ and $\|\mathbf{V}\|_*$. Therefore, it can be solved by using ALM/ADMM method. The corresponding augmented Lagrangian function of the Eqn. (4.11) can be

computed as follows,

$$\begin{aligned} L_\beta(\mathbf{U},\mathbf{V},\mathbf{E},Y) \quad &\triangleq \|\mathbf{W}\odot(\mathbf{X}-\mathbf{E})\|_1+\lambda\|\mathbf{V}\|_*+ \\ &< Y,\mathbf{E}-\mathbf{U}\mathbf{V}>+\frac{\beta}{2}\|\mathbf{E}-\mathbf{U}\mathbf{V}\|_F^2 \end{aligned}$$

where Y is the Lagrange multiplier of linear constraint, β denotes the penalty parameter for the violation of the linear constraint, it is a nonnegative number, $\beta > 0$. The problem (4.11) can be solved using ALM/ADMM approach to minimize the variables iteratively as follows,

$$\begin{cases} \mathbf{U}^{t+1} = \arg\min_{\mathbf{U}} L(\mathbf{U},\mathbf{V}^t,\mathbf{E}^t,Y^t) \\ \mathbf{V}^{t+1} = \arg\min_{\mathbf{V}} L(\mathbf{U}^{t+1},\mathbf{V},\mathbf{E}^t,Y^t) \\ \mathbf{E}^{t+1} = \arg\min_{\mathbf{E}} L(\mathbf{U}^{t+1},\mathbf{V}^{t+1},\mathbf{E},Y^t) \\ Y^{t+1} = Y^t+\beta(\mathbf{E}^{t+1}-\mathbf{U}^{t+1}\mathbf{V}^{t+1}) \end{cases}$$

Given $\mathbf{V}^t$ and $\mathbf{E}^t$, find $\mathbf{U}^{t+1}$

For known $\mathbf{V}^t$ and $\mathbf{E}^t$ in the iteration t, $\mathbf{U}^{t+1}$ can be updated by solving the following equivalent problem,

$$\begin{aligned} \min_{\mathbf{U}} \quad & \frac{\beta}{2}\|(\mathbf{E}^t+\beta^{-1}Y^t)-\mathbf{U}\mathbf{V}^t)\|_F^2 \\ s.t., \quad & \mathbf{U}^\top\mathbf{U}=\mathbf{I} \end{aligned}$$

This is the form of orthogonal Procrustes problem [45]. The global optimal solution can be found using the Singular Value Decomposition approach as follows,

$$[\mathbf{U}',\mathbf{S}',\mathbf{V}'] = SVD((\mathbf{E}^t+\beta^{-1}Y^t)\mathbf{V}^{t\top})$$

or

$$(\mathbf{E}^t+\beta^{-1}Y^t)\mathbf{V}^{t\top} = \mathbf{U}'\mathbf{S}'\mathbf{V}'$$

Finally, $\mathbf{U}^{t+1}$ can be updated as follows [128],

$$\mathbf{U}^{t+1} \leftarrow \mathbf{U}'\mathbf{V}'^{\top}$$

Given $\mathbf{U}^{t+1}$ and $\mathbf{E}^t$, find $\mathbf{V}^{t+1}$

In the second step, when $\mathbf{U}^{t+1}$ and $\mathbf{E}^t$ are given, $\mathbf{V}^{t+1}$ can be found using the following formula,

$$\min_{\mathbf{V}} \lambda\|\mathbf{V}\|_* + < Y^t, \mathbf{E}^t - \mathbf{U}^{t+1}\mathbf{V} > + \frac{\beta}{2}\|\mathbf{E}^t - \mathbf{U}^{t+1}\mathbf{V}\|_F^2$$

Since $\mathbf{U}^{t+1}$ is orthogonal, it can be rewritten as,

$$\min_{\mathbf{V}} \lambda\beta^{-1}\|\mathbf{V}\|_* + \frac{1}{2}\|\mathbf{V} - \mathbf{U}^{t+1\top}(\mathbf{E}^t + \beta^{-1}Y^t)\|_F^2$$

This is the problem of Singular Value Thresholding (SVT) [15]. The soft-thresholding (shrinkage) operator is defined as follows,

$$\mathbf{T}_\tau[x] = max(|x| - \tau, 0)sgn(x)$$

where $sgn(x)$ is the sign function. In SVT, the Singular Value Decomposition is first employed,

$$[\mathbf{U}', \mathbf{S}', \mathbf{V}'] = svd(\mathbf{U}^{t+1\top}(\mathbf{E}^t + \beta^{-1}Y^t))$$

Then, the optimal values of $\mathbf{V}^{t+1}$ can be updated by shrinking the operator $\mathbf{T}_\tau[x]$ to the diagonal matrix $\mathbf{S}'$,

$$\mathbf{V}^{t+1} \leftarrow \mathbf{U}'\mathbf{T}_{\lambda\beta^{-1}}[\mathbf{S}']\mathbf{V}'^{\top}$$

Given $\mathbf{U}^{t+1}$ and $\mathbf{V}^{t+1}$, find $\mathbf{E}^{t+1}$

Given $\mathbf{U}^{t+1}$ and $\mathbf{V}^{t+1}$, $\mathbf{E}^{t+1}$ can be updated using shrinkage technique as follows [128],

$$\min_{\mathbf{E}} \|\mathbf{W} \odot (\mathbf{X} - \mathbf{E})\|_1 + \frac{\beta}{2}\|\mathbf{E} - (\mathbf{U}^{t+1}\mathbf{V}^{t+1} - \beta^{-1}Y^t)\|_F^2$$

Therefore, the observed and missing values in $\mathbf{E}$ can be updated as follows,

$$\begin{cases} \mathbf{W} \odot \mathbf{E} \leftarrow \mathbf{W} \odot (\mathbf{X} - \mathbf{T}_{\beta^{-1}}[\mathbf{X} - \mathbf{U}^{t+1}\mathbf{V}^{t+1} + \beta^{-1}Y^t]) \\ \bar{\mathbf{W}} \odot \mathbf{E} \leftarrow \bar{\mathbf{W}} \odot (\mathbf{U}^{t+1}\mathbf{V}^{t+1} + \beta^{-1}Y^t) \end{cases}$$

where $\bar{\mathbf{W}}$ is the complement of $\mathbf{W}$. Figure 4.5 compares the eigenfaces produced by the regular SVD-ℓ_2 and our SVD-ℓ_1 approach. These eigenfaces are generated using three subjects with frontal faces at 21 different lighting conditions in CMU-PIE database. We can see that the subspaces reconstructed by SVD-ℓ_1 method still keep all properties as the ones in SVD-ℓ_2 method.

4.2.3 Higher-order SVD-ℓ_1

Similar to HOSVD, our proposed Higher-order SVD-ℓ_1 is also a multilinear generalization of ℓ_1-norm based SVD method. Given an N-order tensor $\mathcal{X} \in \mathbb{R}^{d \times n_1 \times \ldots \times n_N}$ that allows missing values and outliers, it can be decomposed using our proposed HOSVD-ℓ_1 into a core tensor $\mathcal{Z}$ and n orthogonal matrices, i.e. a matrix $\mathbf{U}$ and N matrices $\mathbf{V}_i$ to present N factors as follows,

$$\mathcal{X} = \mathcal{Z} \times_1 \mathbf{U}^\top \times_2 \mathbf{V}_1^\top \times_3 \mathbf{V}_2^\top \times_4 \ldots \times_{(N+1)} \mathbf{V}_N^\top \tag{4.12}$$

where $\times_k$ is the *k-mode* matrix product of a tensor. In addition, SVD-ℓ_1 can help to re-formulate HOSVD in the form of Multilinear PCA in combining with Kronecker product.

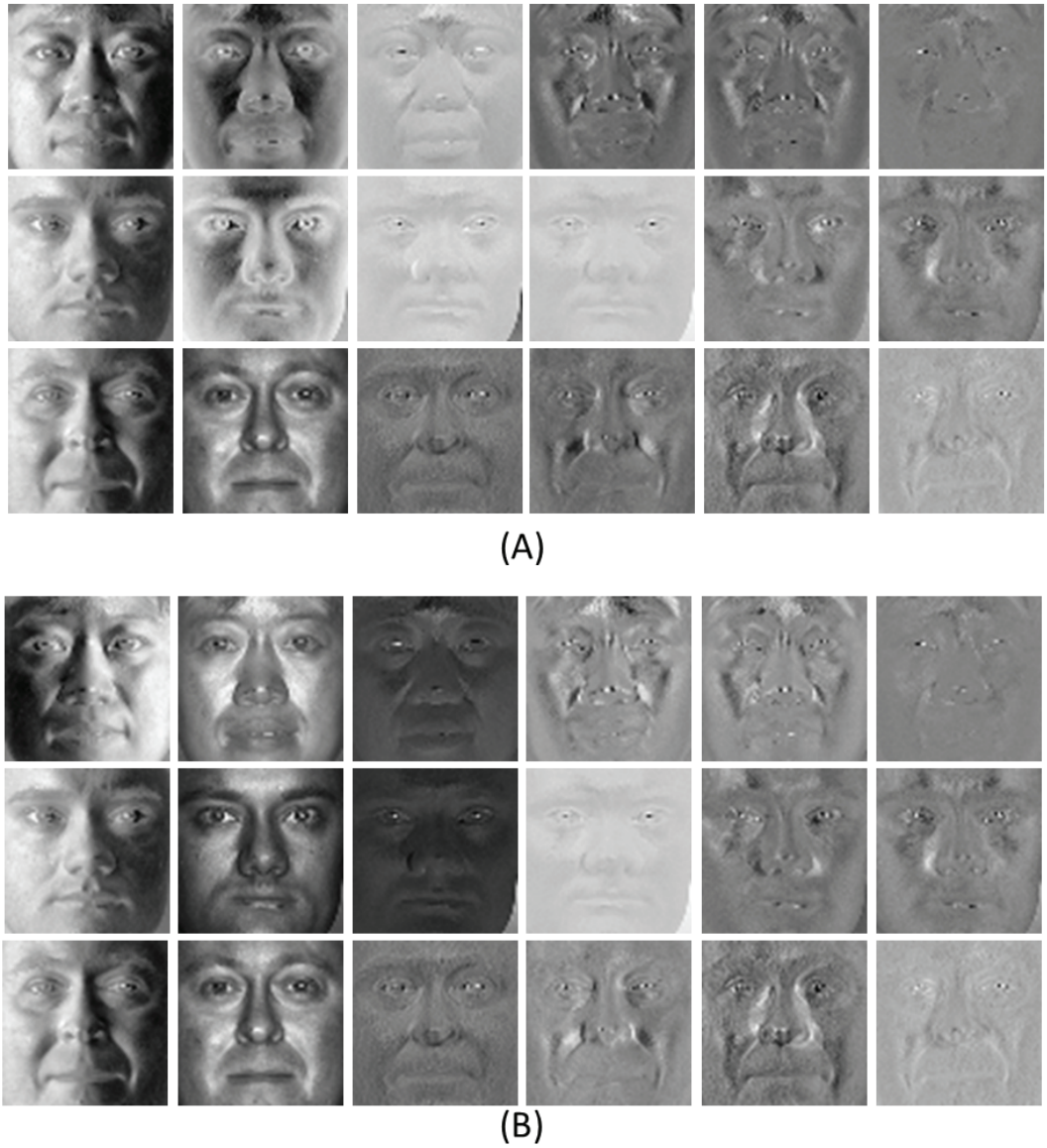


Figure 4.5: Basis eigenvectors produced from CMU-PIE DB. (A) The first six eigenvectors trained by SVD-ℓ_2 on three subjects at frontal pose and 21 different lighting conditions, (B) The corresponding eigenvectors trained by our proposed SVD-ℓ_1 method.

The equivalent form of Eqn. (4.12) in MPCA-ℓ_1 can be denoted as follows,

$$\mathbf{X} = \mathbf{U}\mathbf{Z}(\mathbf{V}_1 \otimes \mathbf{V}_2 \otimes ... \otimes \mathbf{V}_{(N+1)})^\top \tag{4.13}$$

where $\otimes$ denotes the Kronecker product. The matrix $\mathbf{Z}$ results from the pixel-mode flattening of a core tensor. The matrices $\mathbf{U}$ and $\mathbf{V}_k$ are the right singular vector matrix of the

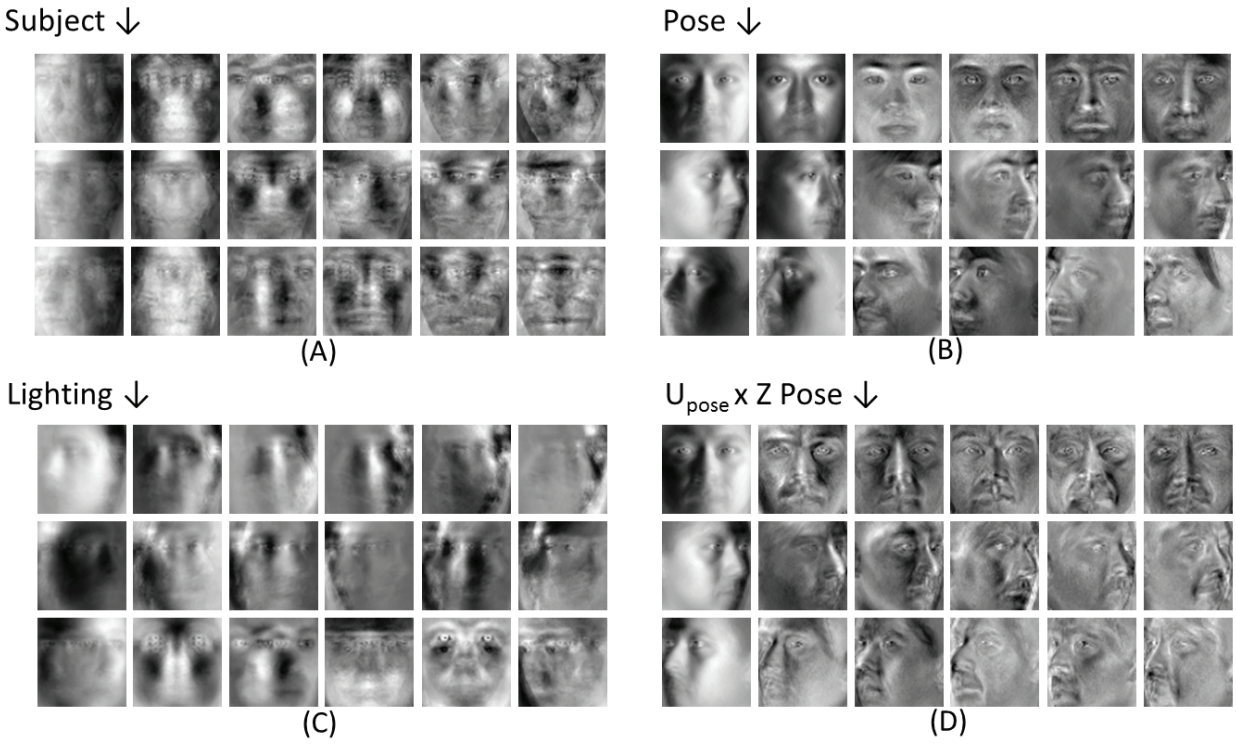


Figure 4.6: Eigenvectors using SVD-ℓ_1 on CMU-PIE: (A) Subject variations, (B) Pose variations, (C) Lighting variations, and (D) $\mathbf{U}_{pose} \times \mathbf{Z}$.

flatten tensor $\mathbf{X}$ along the factor k. They are computed as in Eqn. (4.4). From Eqn. (4.13), $\mathbf{Z}$ can be derived as follows,

$$\mathbf{Z} = \mathbf{U}^\top \mathbf{X}(\mathbf{V}_1 \otimes \mathbf{V}_2 \otimes ... \otimes \mathbf{V}_{(N+1)}) \tag{4.14}$$

Figure 4.6 shows the tensorfaces produced using our SVD-ℓ_1 decomposition approach to the multifactor analysis.

4.3 Higher-order SVD in Random Projection

According to Johnson-Linderstrauss lemma [54], given a set of n points in the Euclidian space $\mathbb{R}^d$, it can be embedded in a high dimension, logarithmic in n and independent of d, without distorting the pairwise distances by more than a factor of $1 \pm \varepsilon$. The idea of

Random Projection has been explained in details in the Johnson-Linderstrauss lemma in chapter 3. In this section, we first present how efficient that low rank approximation is to support in Random Projection. Then the approximation of Random Projection in Singular Value Decomposition is also analyzed in order to guarantee that it can be employed usefully in tensor decomposition.

4.3.1 Low-Rank Approximation

Given a tensor $\mathcal{X}$, there are two steps that can be employed to produce a compressed tensor as follows. First, the tensor $\mathcal{X}$ is projected into a Random Projection subspace with l dimensions in order to get high probability lower dimensions, but is still able to preserve distances and angles of $\mathcal{X}$. Then, the decomposition is employed and preserves the top r singular vectors ($r \leq l$) on the flattened matrices. The low-rank approximation of $\mathcal{X}$ is derived via SVD-ℓ_1 from these components. Alg. 3 shows the main steps in our proposed CSMA method.

4.3.2 Random Projection in SVD

Given a matrix $\mathbf{A}$ with the size of $d \times n$ flattened from a tensor $\mathcal{X}$ along the factor k, let $l \times n$ matrix $\mathbf{B} = \sqrt{\frac{n}{l}}\mathbf{R}^\top\mathbf{A}$ be achieved by embedding $\mathbf{A}$ into a $d \times l$ random matrix $\mathbf{R}$. From Eqn. (2.19), $\mathbf{A}$ and $\mathbf{B}$ can be formulated as follows:

$$\mathbf{A} = \sum_{i=1}^{r_1} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \quad \text{and} \quad \mathbf{B} = \sum_{i=1}^{r_2} \lambda_i \mathbf{a}_i \mathbf{b}_i^T \tag{4.15}$$

We will show that the singular values in Eqn. (4.15) are approximately preserved.

Lemma 2 [111]: Given a positive constant ϵ, if $l \geq C\frac{\log n}{\epsilon^2}$ for sufficiently large constant C then, with high probability,

$$\sum_{i=1}^{r} \lambda_i^2 \geq (1-\epsilon) \sum_{i=1}^{r} \sigma_i^2 \tag{4.16}$$

Algorithm 3 Compressed Submanifold Multifactor Analysis

Input: Tensor $\mathcal{X}$ of facial images with n factors
Output: $\mathbf{U}$, $\mathbf{Z}$ and $\mathbf{V}_k$, $k = 1, ..., n$
Generate RP subspace $\mathbf{R}$ qualified Lemma 1
Low-rank approximation $\mathcal{X}$ to $\tilde{\mathcal{X}}$ via RP subspace $\mathbf{R}$
Flatten $\tilde{\mathcal{X}}$ to $\mathbf{X}$ along the dimension of pixels
$[\mathbf{U}, \mathbf{V}] = \text{SVD-}\ell_1(\mathbf{X})$
for $\forall$ factor $k \in [1..n]$ **do**
  Flatten $\tilde{\mathcal{X}}$ to $\mathbf{X}_k$ along factor k
  $[\mathbf{U}_k, \mathbf{V}_k] = \text{SVD-}\ell_1(\mathbf{X}_k)$
end for
$\mathbf{Z} = \mathbf{U}^\top \times \mathbf{X}_n \times (\mathbf{V}_1 \otimes \mathbf{V}_2 \otimes ... \otimes \mathbf{V}_n)$
Local Coordinate Alignment

Let rank-r approximation to the original matrix be:

$$\tilde{\mathbf{A}}_r = \mathbf{A} \sum_{i=1}^{r} \mathbf{b}_i \mathbf{b}_i^\top \tag{4.17}$$

The main result of this section is presented in the following theorem.

Theorem 1 [111]: For a large enough constant C, if $l \geq C\frac{\log n}{\epsilon^2}$ then

$$\|\mathbf{A} - \tilde{\mathbf{A}}_r\|_F^2 \leq \|\mathbf{A} - \mathbf{A}_r\|_F^2 + 2\epsilon\|\mathbf{A}_r\|_F^2 \tag{4.18}$$

where $\|\mathbf{A}\|_F$ is the Forbenius 2-norm, defined as the sum of squares of all values of $\mathbf{A}$. The difference $\|\mathbf{A} - \mathbf{A}_r\|_F$ presents the accuracy of the approximation. In other words, Theorem 1 shows that the matrix achieved via the two-step approach discussed above gives the same results as the matrix computed via the best rank-r approximation.

4.4 Adaptive Local Coordinate Alignment

In order to avoid the averaging process as in Multilinear PCA, a Stratified approach is employed to generalize procrustes analysis [7, 84]. An example of averaging elimination process can be seen in Figure 4.7. Far apart from other alignment methods, this approach

can deal with missing and noisy data. In this method, we aim to maximize the model likelihood defined as follows:

$$\mathfrak{L}(\mathbf{S}, \mathcal{T}) \triangleq \sigma^{-d}(2\pi)^{-\frac{d}{2}} \prod_{i=1}^{n} \prod_{j=1}^{m} e^{-\frac{\nu_{i,j}}{2\sigma^2} \|\mathbf{A}_{i,j} - \mathbf{T}_i(\mathbf{S}_j)\|_2^2} \tag{4.19}$$

where $\nu_{i,j} \in \{0, 1\}$ is a mask value to represent missing values, $\mathbf{S}^T \triangleq (\mathbf{S}_1, ..., \mathbf{S}_m) \in \mathbb{R}^{d \times m}$, $\mathbf{S}_i \in \mathbb{R}^d$ and $\mathcal{T} \triangleq \{\mathbf{T}_1, ..., \mathbf{T}_n\}$ are unknown reference shape and global transformation respectively. $\mathbf{A}_{i,j} \in \mathbb{R}^d$ are observed samples flattened from tensor $\mathcal{X}$ with a Gaussian distribution of unknown variance σ^2. Maximizing Eqn. (4.19) is equivalent to minimize the data-space cost $\mathcal{E}$:

$$\mathcal{E}(\mathbf{S}, \mathcal{A}) \triangleq \sum_{i=1}^{n} \sum_{j=1}^{m} \nu_{i,j} \|\mathbf{A}_{i,j} - (\mathbf{a}_i \mathbf{S}_j + \mathbf{b}_i)\|_2^2 \tag{4.20}$$

where $\mathcal{T}$ is now the affine transformation $\mathcal{A}$, and $\mathcal{A}_i = (\mathbf{a}_i, \mathbf{b}_i) \in \mathbb{R}^{d \times d} \times \mathbb{R}^d$. The optimal reference shape $\mathbf{S}^T$ is achieved via two steps:

Step 1: Find the initial values by approximating Eqn. (4.20) as follows:

$$\mathcal{E}(\mathbf{S}, \mathcal{A}) \simeq \tilde{\mathcal{E}}(\mathbf{S}, \mathcal{B}) \triangleq \sum_{i=1}^{n} \sum_{j=1}^{m} \nu_{i,j} \|\mathbf{a}_i' \mathbf{A}_{i,j} + \mathbf{b}_i' - S_j\|_2^2 \tag{4.21}$$

where $\mathcal{B}_i = (\mathbf{a}_i', \mathbf{b}_i') \triangleq (\mathbf{a}_i^{-1}, \mathbf{a}_i^{-1}\mathbf{b}_i)$ is the inverse of $\mathcal{A}_i$. It can be interpreted as a negative log likelihood under the hypothesis that the residuals are Gaussian i.i.d. in the registered shape points. The optimization problem is now formed as follows:

$$\min_{\mathbf{S}, \mathcal{B}} \tilde{\mathcal{E}}(\mathbf{S}, \mathcal{B}) \quad s.t. \quad \mathbf{S}^T\mathbf{S} = \mathbf{I} \text{ and } \mathbf{S}^T\mathbf{1} = \mathbf{0} \tag{4.22}$$

The solution $\mathbf{S}^*$ of Eqn. (4.21) can be solved simply using SVD as in [7].

Step 2: Given the initial reference shape which is near-optimal, the iterative Gauss-Newton method is employed to refine the solution.

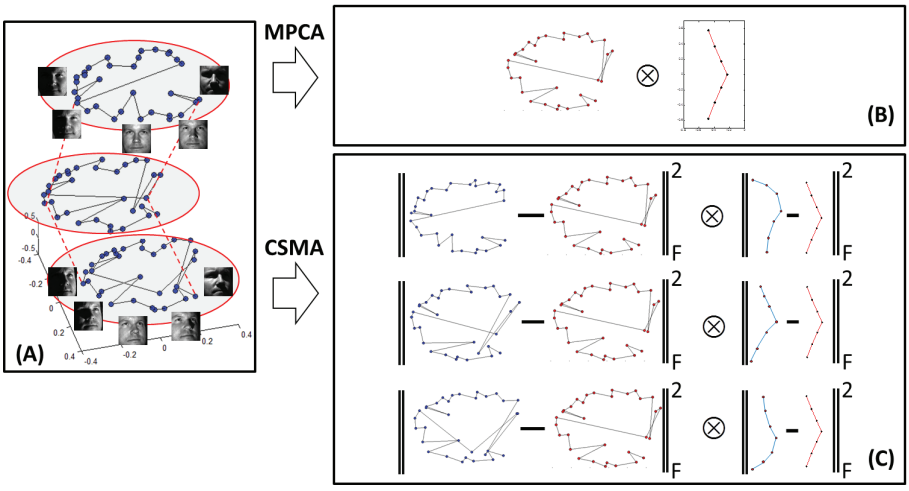


Figure 4.7: A comparison between MPCA and CSMA. Figure (A) is three submanifolds under 30 lighting conditions at 3 different poses from Extended Yale-B database. These submanifolds have different structures. (B) In MPCA decomposition, it aims to preserve the global geometry in data space by averaging all three submanifolds to the same structure. In other words, PCA aims to preserve the distances between all pairs of samples regardless of the presence of multiple factors. Because PCA aims to preserve so much information about all the distances, PCA requires high-dimensional subspaces and does not provide efficient dimension reduction. (C) In CSMA decomposition, it aims to preserve all of the blue and red curves, not merely their averages. Thus, the reconstruction obtained by CSMA more reliably represents the original structure than that obtained by MPCA.

Chapter 5

Sparse Class Dependent Feature Analysis (SCFA)

Dictionary learning methods have recently received a lot of attention for classification problems, to cite a few [5, 53, 115, 120, 126]. While traditional dictionary based approaches like K-Singular Value Decomposition (K-SVD) [5] are optimal for image representation and compression, other methods like Discriminative-KSVD (D-KSVD) [126] and Label Consistent-KSVD (LC-KSVD) [53], incorporate an overcomplete dictionary along with simple discriminative functions in the K-SVD framework, to build optimal classifiers. Due to this simplistic classification structure, their performance is usually limited. In this chapter, we propose a novel approach named Sparse Class-dependent Feature Analysis (SCFA), to combine the advantages of sparse representation in an overcomplete dictionary, with a powerful nonlinear classifier. The classifier is based on the estimation of class-specific optimal filters, by solving an ℓ_1-norm optimization problem. We show how this problem is solved by using the Alternating Direction Method of Multipliers and also explore relevant convergency details. Our method as well as its Reproducing Kernel Hilbert Space (RKHS) version is tolerant to the presence of noise and other variations in the image. We achieved very high classification accuracies in face recognition (in the presence of lighting

variations, i.e. Extended Yale database [44] and with occlusions, i.e. AR database [75]) and object (Caltech101) [41] databases. Our method outperforms the state of the art in all these databases and hence shows its applicability to general computer vision and pattern recognition problems.

5.1 Dictionary Learning Based for Classification

Sparse representation in overcomplete dictionaries has been gaining a lot of popularity in recent years, with various studies being published, regarding their utility in image denoising [36], compression [14], classification [115] amongst others. The use of such overcomplete "basis functions" for signal representation has advantages such as flexibility, robustness to noise, etc. These methods 'infer' a dictionary from a set of exemplary samples. The Method of Optimal Directions (MOD) [59] and K-Singular Value Decomposition (K-SVD) [5] are two examples of this class of methods.

These algorithms have garnered significant attention recently, for the purpose of building dictionaries for signal classification. Yang *et al* [120] developed a unified framework to learn both a classification model and a corresponding overcomplete dictionary. Other approaches such as the D-KSVD approach by Zhang and Li [126] incorporated a dictionary based classification function into the K-SVD framework and thus jointly solved an optimization problem for dictionary building. A more recent paper by Jiang *et al* [53] called the Label Consistent KSVD (LC-KSVD) incorporated a strategy of 'class specific dictionary atoms' into their K-SVD framework. Their work also contains a function to minimize the classification error based on these dictionary atoms. Other algorithms like [72, 73] treat dictionary learning and classifier design as two separate problems and do not derive a single dictionary to solve both problems simultaneously. The advantage of decoupling these two design requirements is that the classification design becomes simple allowing the incorporation of powerful nonlinear classification tools such as the Kernel Class-dependent

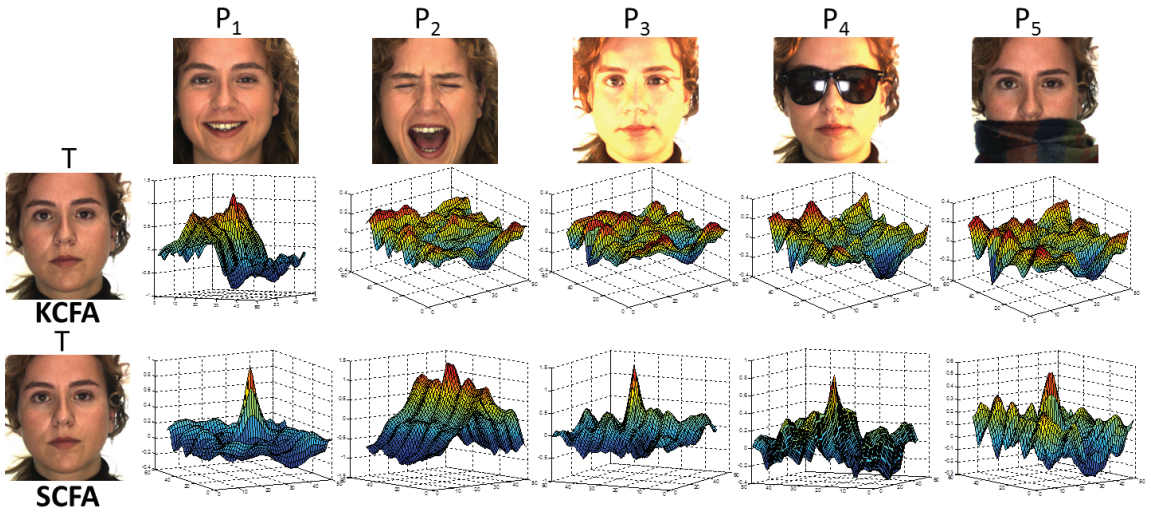


Figure 5.1: A comparison between KCFA and SCFA for face matching on AR face database. Given probe images P_i with different variations, e.g. facial expressions, lighting and occlusions, not in the target images T, the filter responses in SCFA to the correct target subject are usually sharper and stronger than the ones in KCFA.

Feature Analysis (KCFA) approach [93, 117]. This has been used successfully for both classification and dimensionality reduction. However, KCFA by itself is very sensitive to outliers and noise in image data. Furthermore, it uses raw image pixel values as inputs during the design phase. Hence its application is limited to matching images acquired under heavy constraints. Its performance drops dramatically when used with face databases having a lot of appearance variation, human iris data or with general objects presented in the Caltech101 dataset, due to the presence of a lot of variability therein. The first row in Figure 5.1 with very low filter responses shows an example of that problem.

In this chapter, we propose a novel dictionary learning based pattern classification method named Sparse Class-dependent Feature Analysis (SCFA) to overcome the limitations discussed above. Our method is a highly discriminative, dictionary based classification approach using optimal image filters. The filter design is set up as an ℓ_1-norm based optimization problem, which is solved using the Alternating Direction Method of Multipliers (ADMM). The performance reported here, on a variety of standard databases shows that our method surpasses the state-of-the-art and is applicable to general computer vision

and pattern recognition problems as shown in Figure 5.2.

The remainder of this chapter is organized as follows. In section 5.2, we briefly review the background as well as the solutions in the regular Class Feature-dependent Analysis and its kernel version. We first review the fundamental ideas of the correlation filters and class-dependent feature analysis. Then, we show how to find the solution by using the regular ℓ_2-norm formulation. Its kernel version that allows achieving better accuracy rates is also presented in this section. In section 5.3, we introduce our proposed Sparse Class-dependent Feature Analysis method solved in the form of ℓ_1-norm optimization. Its advanced version implemented in the Reproducing Kernel Hilbert Space that allows the tolerant capability to the presence of noise and other variations from given images is also presented in this section.

5.2 Kernel Class-dependent Feature Analysis

The method of Class-dependent Feature Analysis [93] uses a set of Minimum Average Correlation Energy (MACE) filters [71] to extract facial features. For classifying a given testing image, MACE filters are determined for every subject during the training stage. We briefly review this method in this section.

5.2.1 Class-dependent Feature Analysis (CFA)

Given N training images of size $M \times M$, the Average Correlation Energy (ACE) of a filter response [71] can be computed as follows,

$$\begin{aligned} E &= \frac{1}{d.N} \sum_{i=1}^{N} \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} |H(u,v)|^2 |X_i(u,v)|^2 \\ &= \frac{1}{d.N} \sum_{i=1}^{N} (\mathbf{h}^+ \mathbf{X}_i)(\mathbf{X}_i^+ \mathbf{h}) \\ &= \mathbf{h}^+ \mathbf{D} \mathbf{h} \end{aligned} \tag{5.1}$$

where $d = M^2$, $H(u, v)$ is the filter in frequency domain and $X_i(u, v)$ is the 2D Fourier transform of the ith training image. $\mathbf{D} = \frac{1}{d.N} \sum_{i=1}^{N} \mathbf{X}_i \mathbf{X}_i^+$ is a $d \times d$ diagonal matrix with the average power spectrum of the training images along its diagonal, and $+$ indicates the complex conjugate transpose of a matrix. The MACE filter pre-specifies the value of correlation peak at origin given by,

$$\mathbf{X}^+ \mathbf{h} = \mathbf{u} \tag{5.2}$$

where $\mathbf{X}$ is a $d \times N$ complex valued matrix. The column vector $\mathbf{u}$ contains pre-specified correlation values, with 1 at the origin, for the 'authentic' class to which the MACE filter corresponds and 0 for all other images belonging to the 'imposter' classes.

5.2.2 CFA Solution Analysis

The CFA solution is obtained by minimizing the energy E subject to the constraint (5.2) and can be reformulated,

$$\mathbf{h}^* \triangleq \arg\min_h \|\mathbf{h}^+ \mathbf{D} \mathbf{h}\|_2 \quad \text{s.t.} \quad \mathbf{u} = \mathbf{X}^+ \mathbf{h} \tag{5.3}$$

The Lagrangian form to solve (5.3) is given by,

$$\mathbf{\Gamma}(\mathbf{h}) \triangleq \|\mathbf{h}^+ \mathbf{D} \mathbf{h}\|_2 + \boldsymbol{\lambda}^T (\mathbf{X}^+ \mathbf{h} - \mathbf{u}) \tag{5.4}$$

where $\boldsymbol{\lambda}$ are the Lagrange multipliers. Setting the derivative of $\mathbf{\Gamma}(\mathbf{h})$ w.r.t. $\mathbf{h}$ to zero, gives the optimal solution,

$$\frac{\partial \mathbf{\Gamma}(\mathbf{h})}{\partial \mathbf{h}} = 2\mathbf{D}\mathbf{h} + \boldsymbol{\lambda}^T \mathbf{X}^+ = 0 \tag{5.5}$$

From (5.3) and (5.5), we can derive,

$$\mathbf{u} = \mathbf{X}^+ \mathbf{h}^* = -\frac{1}{2} \mathbf{X} \mathbf{D}^{-1} \mathbf{X}^+ \boldsymbol{\lambda} \tag{5.6}$$

This is the closed-form solution, where $\mathbf{X}$ is a full-rank matrix. Hence, the MACE filter $\mathbf{h}^*$ can be derived,

$$\mathbf{h}^* = \mathbf{D}^{-1}\mathbf{X}(\mathbf{X}^+\mathbf{D}^{-1}\mathbf{X})^{-1}\mathbf{u} \tag{5.7}$$

In practice, the Optimal Tradeoff Synthetic Discriminant Function (OTSDF) filter [86] is used, instead of the MACE filter, since it has noise tolerance $\mathbf{C}$. The OTSDF is designed to minimize the output noise variance $\mathbf{h}^+\mathbf{C}\mathbf{h}$ while holding the average correlation energy $\mathbf{h}^+\mathbf{D}\mathbf{h}$ fixed. Similar to Eqn. (5.7), the final OTSDF filter can be shown to be,

$$\mathbf{h}^* = \mathbf{T}^{-1}\mathbf{X}(\mathbf{X}^+\mathbf{T}^{-1}\mathbf{X})^{-1}\mathbf{u} \tag{5.8}$$

where $\mathbf{T} = (\alpha\mathbf{D} + \sqrt{1-\alpha^2}\mathbf{C})$ and $0 \leq \alpha \leq 1$ is a parameter that controls the trade-off between noise tolerance and discrimination. In our experiments, α is empirically set to 0.3 which achieves the high recognition accuracy.

Given a test image $\mathbf{y}$, the correlation of that image with N OTSDF filters can be expressed as,

$$\mathbf{c}^* = \mathbf{H}^\top\mathbf{y} = [\mathbf{h}_1\mathbf{h}_2\cdots\mathbf{h}_N]^\top\mathbf{y} \tag{5.9}$$

where $\mathbf{h}_i$ is an OTSDF filter corresponding to class *i*. The assigned class is chosen based on the highest filtered response value.

5.2.3 Kernel Class-dependent Feature Analysis (KCFA)

Kernel CFA overcomes the poor performance of the linear CFA classifier due to the presence of nonlinearities in the data. Data is 'pre-whitened' before deriving the kernel OTSDF filter closed form solution. The correlation output of test image $\mathbf{y}$ with filter $\mathbf{h}$, using

Eqn. (5.8) is,

$$\begin{aligned}\mathbf{y}^{+}\mathbf{h} &= \mathbf{y}^{+}[\mathbf{T}^{-1}\mathbf{X}(\mathbf{X}^{+}\mathbf{T}^{-1}\mathbf{X})^{-1}\mathbf{u}] \\ &= \left((\widetilde{\mathbf{y}})^{+}\widetilde{\mathbf{X}}\right)\left((\widetilde{\mathbf{X}})^{+}\widetilde{\mathbf{X}}\right)^{-1}\mathbf{u}\end{aligned} \tag{5.10}$$

where $\widetilde{\mathbf{X}} = \mathbf{T}^{-\frac{1}{2}}\mathbf{X}$ and $\widetilde{\mathbf{y}} = \mathbf{T}^{-\frac{1}{2}}\mathbf{y}$ denote the pre-whitened version of $\mathbf{X}$ and $\mathbf{y}$. Using the kernel trick here yields the kernel correlation filter with mapping $\mathbf{\Phi}$,

$$\begin{aligned}\mathbf{\Phi}(\mathbf{y}) \cdot \mathbf{\Phi}(\mathbf{h}) &= \left(\mathbf{\Phi}(\mathbf{y}) \cdot \mathbf{\Phi}(\mathbf{X})\right)\left(\mathbf{\Phi}(\mathbf{X}) \cdot \mathbf{\Phi}(\mathbf{X})\right)^{-1}\mathbf{u} \\ &= K(\mathbf{y}, \mathbf{x}_i)K(\mathbf{x}_i, \mathbf{x}_j)^{-1}\mathbf{u}\end{aligned}$$

Although KCFA has been used successfully in face recognition [93, 117], it suffers from major drawbacks. First, it is very sensitive to outliers. The performance drops dramatically when data contains noise or outliers, due to illumination variation, noise during acquisition etc. as shown in Figure 5.1. Second, it is also sensitive to variations in image appearance, e.g. facial expressions, pose variations, object perspective, etc. Hence, it is applied only to classification problems with a fixed template, i.e. frontal faces, fixed view target recognition etc. The reason for this drawback is that, OTSDF filters are trained using image pixel values. The proposed SCFA method, addresses these drawbacks by using sparsity in an overcomplete dictionary, in the filter design phase.

5.3 Sparse Class-dependence Feature Analysis (SCFA)

This section describes our dictionary based SCFA classifier, which outperforms state-of-the-art discriminative dictionary based classifiers, while overcoming the limitations of KCFA classifiers discussed above. This method involves estimating the filters $\mathbf{h}$ in Eqn.(5.7) by solving a constrained ℓ_1-norm problem. The objective function is converted into an equivalent convex problem, then put into an Alternating Direction Method of Multipliers

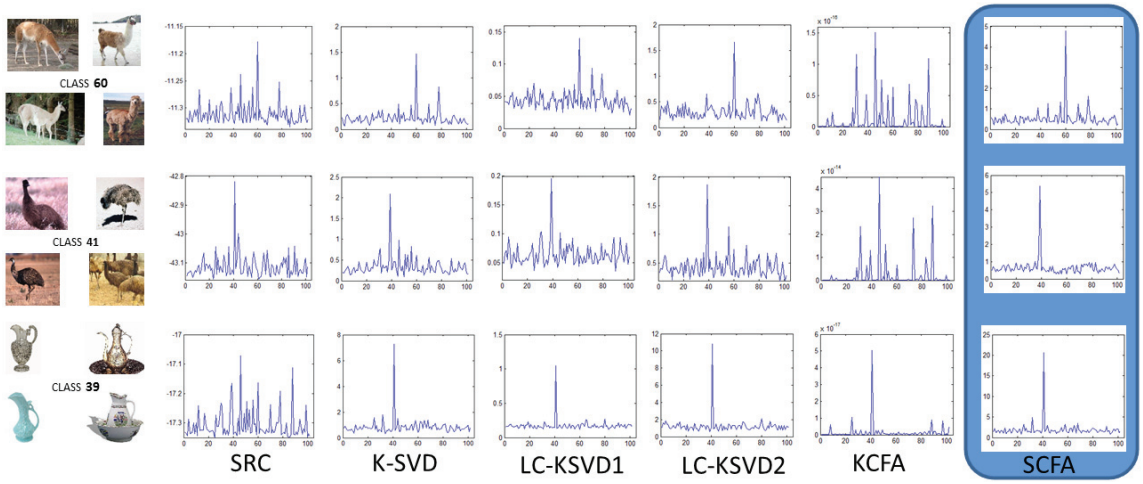


Figure 5.2: An example to show the discriminative power of SCFA compared to state of the art. The sum of all classification peak values corresponding to subject 60, 41 and 39 from the Caltech101 database are shown in row 1, 2 and 3 respectively using various methods. SCFA hardly shows any response for classes other than the 'genuine' class. All other methods show responses for 'imposter' classes too.

(ADMM) framework. Our solution to this problem as well as stopping criteria are presented in this chapter.

5.3.1 ℓ_1-norm Filter Design

Consider the constraint in Eqn. (5.3) where the data matrix $\mathbf{X}$ is expressed in terms of its pre-whitened form $\widetilde{\mathbf{X}} = \mathbf{T}^{-\frac{1}{2}}\mathbf{X}$ (Eqn. (5.10)), the following constraint can be derived,

$$\mathbf{u} = (\mathbf{T}^{\frac{1}{2}}\widetilde{\mathbf{X}})^{+}\mathbf{h} = \widetilde{\mathbf{X}}^{+}(\mathbf{T}^{\frac{1}{2}}\mathbf{h}) = \widetilde{\mathbf{X}}^{+}\widetilde{\mathbf{h}} \tag{5.11}$$

Similarly, the objective function term,

$$\begin{aligned} \mathbf{h}^{*} &\triangleq \arg\min_{\mathbf{h}} \|\mathbf{h}^{+}\mathbf{T}\mathbf{h}\|_2 \\ &= \arg\min_{\mathbf{h}} \|(\mathbf{T}^{\frac{1}{2}}\mathbf{h})^{+}(\mathbf{T}^{\frac{1}{2}}\mathbf{h})\|_2 \\ &= \arg\min_{\mathbf{h}} \|\tilde{\mathbf{h}}^{+}\tilde{\mathbf{h}}\|_2 \quad \text{s.t. } \mathbf{u} = \widetilde{\mathbf{X}}^{+}\widetilde{\mathbf{h}} \end{aligned} \tag{5.12}$$

Hence the equivalent problem to (5.3) can be expressed as,

$$\mathbf{h}^* \triangleq \arg\min_{\widetilde{\mathbf{h}}} \|\widetilde{\mathbf{h}}\|_2^2 \quad \text{s.t. } \mathbf{u} = \widetilde{\mathbf{X}}^+ \widetilde{\mathbf{h}} \tag{5.13}$$

Instead of the ℓ_2-norm used here, the problem will be solved using ℓ_1-norm, which estimates a sparse solution. In the following analysis, $\mathbf{h}$ is used in place of $\widetilde{\mathbf{h}}$ for simplicity of notation. The redefined problem is as follows,

$$\mathbf{h}^* \triangleq \arg\min_{\mathbf{h}} \|\mathbf{F}\mathbf{h}\|_1 \quad \text{s.t. } \mathbf{u} = \widetilde{\mathbf{X}}^+ \mathbf{h} \tag{5.14}$$

where $\mathbf{F}$ can be an arbitrary form with 'weights' along the diagonal to emphasize important components in $\mathbf{h}$ based on prior information. This problem is traditionally solved using a LASSO solver [24],

$$\min_{\mathbf{h}} \frac{1}{2}\|\widetilde{\mathbf{X}}^\top \mathbf{h} - \mathbf{u}\|_2^2 + \lambda \|\mathbf{F}\mathbf{h}\|_1 \tag{5.15}$$

where λ is a parameter that controls the trade-off between sparsity and reconstruction fidelity. Several algorithms can be found in literature to solve this problem, i.e. homotopy [76], LARS [34].

In our work, this problem is solved in the ADMM framework [13] which is easy to compute and converges fast at the early stage. Eqn. (5.15) can easily be converted into this framework by introducing a constraint vector $\mathbf{z}$. Then, it can be re-written with an explicit constraint as follows,

$$\begin{aligned} &\min_{\mathbf{h}} \frac{1}{2}\|\widetilde{\mathbf{X}}^\top \mathbf{h} - \mathbf{u}\|_2^2 + \lambda \|\mathbf{z}\|_1 \\ &\qquad s.t. \quad \mathbf{F}\mathbf{h} - \mathbf{z} = 0 \end{aligned} \tag{5.16}$$

The corresponding *augmented Lagrangian* form for Eqn. (5.16) is given below,

$$L_\rho(\mathbf{h}, \mathbf{z}, \mathbf{y}) = \frac{1}{2}\|\widetilde{\mathbf{X}}^\top \mathbf{h} - \mathbf{u}\|_2^2 + \lambda\|\mathbf{z}\|_1 + \\ \rho\mathbf{y}^\top(\mathbf{F}\mathbf{h} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{F}\mathbf{h} - \mathbf{z}\|^2 \tag{5.17}$$

where $\mathbf{y}$ is the augmented Lagrange multipliers and $\rho > 0$ is the step length. Eqn. (5.16) can be then optimally solved using the following iterative scheme. In the following expressions, the superscript 't' indicates the t-th iteration.

1) Solve for $\mathbf{h}^{t+1}$ with fixed $\mathbf{z}^t$ and $\mathbf{y}^t$

This is done simply by setting the derivative of Eqn. (5.17) w.r.t. $\mathbf{h}$ to zero and solving for $\mathbf{h}$,

$$\mathbf{h}^{t+1} = (\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top + \rho\mathbf{F}^\top\mathbf{F})^{-1}(\widetilde{\mathbf{X}}\mathbf{u} + \rho\mathbf{F}^\top(\mathbf{z}^t - \mathbf{y}^t)) \tag{5.18}$$

2) Solve for $\mathbf{z}^{t+1}$ with fixed $\mathbf{h}^{t+1}$ and $\mathbf{y}^t$

In Eqn. (5.17), when $L_\rho(\mathbf{h}, \mathbf{z}, \mathbf{y})$ is minimized according to $\mathbf{z}$, it then becomes,

$$\mathbf{z}^{t+1} = \arg\min_{\mathbf{z}} \lambda\|\mathbf{z}\|_1 + \rho\mathbf{y}^{t\top}(\mathbf{F}\mathbf{h}^{t+1} - \mathbf{z}) + \frac{\rho}{2}\|\mathbf{F}\mathbf{h}^{t+1} - \mathbf{z}\|^2 \tag{5.19}$$

This problem can be solved via a simple soft thresholding operator [31],

$$\mathbf{z}^{t+1} = S_{\lambda/\rho}(\mathbf{F}\mathbf{h}^{t+1} + \mathbf{y}^t) \tag{5.20}$$

For group Lasso ($\lambda\sum_i \|x_i\|_2$ for disjoint $x_i \in \mathbb{R}^{n_i}$), the soft thresholding operator can be defined as follows,

$$S_\kappa(a) = \left(1 - \frac{\kappa}{\|a\|_2}\right)_+ a \tag{5.21}$$

where $(.)_+$ is the positive part.

3) Solve for $\mathbf{y}^{t+1}$ with fixed $\mathbf{h}^{t+1}$ and $\mathbf{z}^{t+1}$

In the final step, the Lagrange multiplier $\mathbf{y}^{t+1}$ can be updated as follows [33],

$$\mathbf{y}^{t+1} = \mathbf{y}^t + \mathbf{F}\mathbf{h}^{t+1} - \mathbf{z}^{t+1} \tag{5.22}$$

5.3.2 Stopping Criteria

The convergence in ADMM has been studied in several previous work [13, 33]. Results were shown with general penalties or inexact minimization via Douglas-Rachford splitting, splitting operator, etc. In the primal feasibility problem for general ADMM, the necessary and sufficient optimality conditions for the optimal value p^* is as follows,

$$\mathbf{A}\mathbf{h}^* + \mathbf{B}\mathbf{z}^* - c = 0 \tag{5.23}$$

where $\mathbf{A} = \mathbf{F}$, $\mathbf{B} = -\mathbf{1}$ and $c = 0$ in the case of Eqn. (5.16), and the corresponding dual feasibility is,

$$\begin{aligned} 0 &\in \partial f(\mathbf{h}^*) + \mathbf{A}^\top \mathbf{y}^* \\ 0 &\in \partial g(\mathbf{z}^*) + \mathbf{B}^\top \mathbf{y}^* \end{aligned} \tag{5.24}$$

where $f(\mathbf{h}) = \|\widetilde{\mathbf{X}}^\top \mathbf{h} - \mathbf{u}\|_2^2$, $g(\mathbf{z}) = \|\mathbf{z}\|_1$ and ∂ is the subdifferential operator. The residuals for the primal and dual feasibility condition at iteration k can be computed as $r^k = \mathbf{A}\mathbf{h}^k + \mathbf{B}\mathbf{z}^k - c$ and $s^k = \rho \mathbf{A}^\top \mathbf{B}(\mathbf{z}^k - \mathbf{z}^{k-1})$.

The residuals of the optimality conditions can be related to a bound on the objective sub-optimality of the current point. From the convergence proof in [13], we can derive,

$$\begin{aligned} f(\mathbf{x}^k) + g(\mathbf{z}^k) - p^* &\leq -(\mathbf{y}^k)^\top r^k + d\|s^k\|_2 \\ &\leq \|\mathbf{y}^k\|_2 \|r^k\|_2 + d\|s^k\|_2 \end{aligned} \tag{5.25}$$

where $\|\mathbf{h}^k - \mathbf{h}^*\|_2 \leq d$. Therefore, the reasonable termination criterion is that the primal residual r^k and dual one s^k have to be small enough,

$$\|r^k\|_2 \leq \sqrt{p}\epsilon^{abs} + \epsilon^{rel} \max\{\|\mathbf{A}\mathbf{h}^k\|_2, \|\mathbf{B}\mathbf{z}^k\|_2, \|c\|_2\}$$
$$\|s^k\|_2 \leq \sqrt{n}\epsilon^{abs} + \epsilon^{rel}\|\mathbf{A}^\top \mathbf{y}^k\|_2$$

where $\epsilon^{abs}, \epsilon^{rel} > 0$ denote the absolute and relative tolerance respectively. The factors $\sqrt{p}$ and $\sqrt{n}$ are computed due to the ℓ_2 norms are in R^p and R^n respectively. In our case, the relative stopping criterion is set $\epsilon^{rel} = 10^{-3}$.

5.3.3 Discriminative Dictionary for Sparse Coefficients

In the classical KCFA method, pixel values are used as inputs. Therefore, it does not have the capability to represent images robustly. It is easily affected by noise distortions and illumination variations. Instead, in our method, sparse coefficients extracted from a discriminative dictionary are used as inputs to the filters discussed in 5.3.1.

Given a set $\mathbf{Y}$ of d-dimensional N images, $\mathbf{Y} \in \mathbb{R}^{d \times N}$, the overcomplete dictionary $\mathbf{\Upsilon}$ ($\mathbf{\Upsilon} = [\upsilon_1, ..., \upsilon_K] \in \mathbb{R}^{d \times K}$) with K items ($K > d$) for sparse representation $\mathbf{X}$ ($\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N] \in R^{K \times N}$) of $\mathbf{Y}$ can be achieved via the following optimization problem,

$$< \mathbf{\Upsilon}, \mathbf{X} >= \arg\min_{\mathbf{\Upsilon}, \mathbf{X}} \|\mathbf{Y} - \mathbf{\Upsilon}\mathbf{X}\|_2^2 \quad s.t. \quad \forall i, \|\mathbf{x}_i\|_0 \leq \xi \tag{5.26}$$

where $\|\mathbf{Y} - \mathbf{\Upsilon}\mathbf{X}\|_2^2$ denotes the reconstruction error and ξ is a sparsity constraint factor to guarantee that each signal has fewer than ξ items in its decomposition.

The overcomplete dictionary $\mathbf{\Upsilon}$ and $\mathbf{X}$ can be found from Eqn. (5.26) using the K-SVD algorithm. This approach works robustly and efficiently in the applications of image compression and representation. Given a trained overcomplete dictionary $\mathbf{\Upsilon}$, sparse coding

computes the sparse representation $\mathbf{X}_{Test}$ of a testing set $\mathbf{Y}_{Test}$ by solving,

$$\mathbf{X}_{Test} = \arg\min_{\mathbf{X}} \|\mathbf{Y}_{Test} - \boldsymbol{\Upsilon}\mathbf{X}\|_2^2 \quad s.t. \quad \forall i, \|\mathbf{x}_i\|_0 \leq \xi$$

5.3.4 Reproducing Kernel Hilbert Space (RKHS)

The filter design in section 5.3.1 can be extended to an RKHS. Eqn. (5.14) can be rewritten in a kernel version as,

$$\mathbf{h}^* \triangleq \arg\min_{\mathbf{h}} \|\mathbf{F}\mathbf{h}\|_1 \quad \text{s.t. } \mathbf{u} = \mathbf{X}_K^+\mathbf{h} \tag{5.27}$$

where $\mathbf{X}_K$ are mapped from the original $\mathbf{X}$ using a kernel mapping $\boldsymbol{\Phi}$. $\boldsymbol{\Phi} : \mathbb{R}^N \mapsto F$, and the Radial Basis Function (RBF) kernel is defined by,

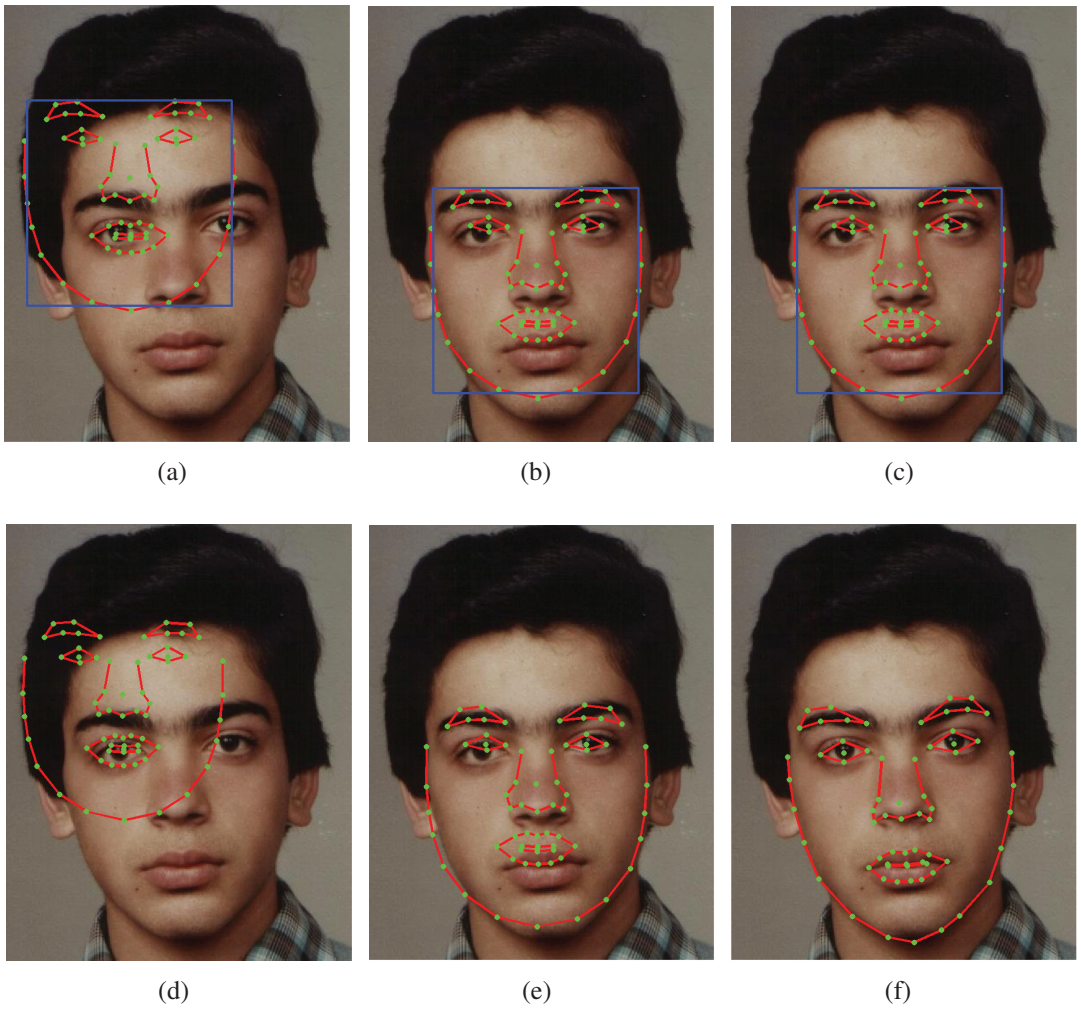
$$K(\mathbf{x}, \mathbf{y}) = \langle \boldsymbol{\Phi}(\mathbf{x}), \boldsymbol{\Phi}(\mathbf{y}) \rangle = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$$

In our experiments, SCFA in RKHS always reports higher accuracy compared to the linear one. Therefore, all our experiments in section 6.4 are reported using SCFA in RKHS.

Chapter 6

Experimental Results

In this chapter, we first provide a detailed analysis of the usefulness of our proposed Compressed Submanifold Multifactor Analysis on both random or toy samples and challenging face databases. We demonstrate that our CSMA method can outperform the other state-of-the-art methods in classification results. In addition, we show that our method supports the efficient functionalities that we cannot find in the other classical tensor decomposition methods. Then, in the second section of this chapter, we show that our Sparse Class Dependent Feature Analysis achieves highest classification rates compared to the other sparse-based classification methods in different modalities, e.g. objects and faces. The facial images collected from the databases are detected and landmarked automatically with 79 points using Modified Active Shape Model (MASM) [94, 95] which allows to give higher landmarking accuracy than the classical Active Shape Model (ASM) method [25]. Figure 6.1 shows an example of MASM that gives better fitting results than the classical ASM method.



(a) (b) (c)

(d) (e) (f)

Figure 6.1: Sample ASM fitting results. The images in the first row are the initialization provided to the ASMs while the images in the second row show the corresponding fitting results under such initialization conditions. (a) An example of poor initialization, (b) Accurate initialization provided to the classical ASM implementation (c) Accurate initialization provided to MASM, (d) Fitting results produced by MASM under poor initialization conditions, (e) Fitting results produced by classical ASM under accurate initialization conditions, (f) Fitting results produced by MASM under accurate initialization conditions.

6.1 CSMA Experiments

6.1.1 CSMA in Tensors with Random Values

In this section, in order to show the robustness of our proposed Compressed Submanifold Multifactor Analysis, it will be evaluated on tensors constructed by random values. We first

Table 6.1: CSMA Reconstruction Errors on Tensors with Missing Values.

% of missing data	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Mean	0.0	0.0	16.2	37.4	58.8	75.1	94.9	113.3	135.5	150
SD	0.0	0.0	0.55	4.9	8.5	5.5	1.9	21.7	31.1	28.8

show its ability in decomposing tensors with missing values. Then, in the second experiment, we will show the robustness of our proposed method against tensors with random noise values.

CSMA on Tensors with Missing Values

In the first experiment, in order to show the robustness against missing values, our proposed CSMA method is evaluated on random or toy samples with missing values. A 3-factors tensor $\mathcal{X} \in \mathbb{R}^{10\times15\times20\times10}$, where the first dimension presents pixels, is randomly generated with a set of missing values. Unfortunately, Principal Component Analysis [55, 105], Linear Discriminant Analysis [8], Linearity Preserving Projection [51], Multilinear PCA and Submanifold Preserving Multifactor Analysis cannot be employed in this case due to the missing data. However, our proposed Compressed Submanifold Multifactor Analysis approach is able to solve this problem efficiently. The tensor $\mathcal{X}$ is decomposed into subspace $\mathbf{U}$, core matrix $\mathbf{Z}$ and three factor matrices $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$, where $\tilde{\mathcal{X}} = \mathbf{UZ}(\mathbf{V}_1 \otimes \mathbf{V}_2 \otimes \mathbf{V}_3)^\top$. It is generated with a different number of missing values varying from 0% to 90% of its sizes. For each number of missing values, the experiments are repeated $N = 50$ times. The mean $1/N \sum_{i=1}^{N} \|\mathcal{X} - \tilde{\mathcal{X}}_i\|$ and the standard deviation (SD) of the reconstruction errors are computed. Table 6.1 shows the reconstruction errors with different percentages of missing values. This table shows the efficiency of our CSMA method in reconstructing tensor $\tilde{\mathcal{X}}$ from a given tensor $\mathcal{X}$ with missing data. Even for large percentages of missing data (up to 70%), the reconstruction errors are reasonable and the standard deviations are small. However, above 70% of missing data, the standard deviation increases significantly.

Table 6.2: CSMA Reconstruction Errors (PSNR) on Tensors with Noisy Values (Mean $\pm$ SD).

Noise (σ)	PCA	MPCA	SPMA	CSMA
0.0	295 ± 1.3	300 ± 6.9	59 ± 4.7	351 ± 0.5
0.1	20 ± 1.7	20 ± 2.1	19 ± 1.5	25 ± 0.5
0.2	15 ± 3.4	16 ± 1.3	16 ± 6.2	19 ± 4.1
0.3	13 ± 3.4	15 ± 1.3	15 ± 6.2	18 ± 4.1

CSMA on Tensors with Noise

In the second experiment, in order to show the robustness against noise, our CSMA method is evaluated on a 3-factors tensor $\mathcal{X} \in \mathbb{R}^{10\times15\times20\times10}$. This $\mathcal{X}$ is randomly generated with additive white Gaussian noise with the mean of zero and the standard deviation σ varying from 0.0 to 0.9. It is then decomposed using four different methods, i.e. PCA, MPCA, SPMA and our proposed CSMA approach. The reconstruction errors between the original $\mathcal{X}$ and the reconstructed $\tilde{\mathcal{X}}$ are computed using the Peak Signal-to-Nose Ratio (PSNR) scores (decibels) as shown in Eqn. (6.1).

$$PSNR = 20 \times \log_{10}\left(\frac{\max(\mathcal{X})}{\sqrt{MSE}}\right) \tag{6.1}$$

where $\max(\mathcal{X})$ is the maximum positive value in the tensor $\mathcal{X}$ and MSE is the mean square error between the original tensor $\mathcal{X}$ and the reconstructed one $\tilde{\mathcal{X}}$ with the sizes of $n_1 \times n_2 \times n_3 \times n_4$:

$$MSE = \frac{1}{\Pi_{i=1}^{4} n_i}\|\mathcal{X} - \tilde{\mathcal{X}}\|_2^2 \tag{6.2}$$

This evaluation scheme is also repeated $N = 50$ times at every given σ value. The mean and the SD of the PSNR scores at these σ values are collected. The experimental results are shown in Table 6.2. Our proposed CSMA approach gives the highest PSNR scores in the reconstruction among four decomposition methods.

6.1.2 The Robustness of Random Projection

In order to study the effectiveness of random projection, our proposed system is evaluated with different sizes of the random projection subspace constructed on CMU-PIE face database [96]. This database includes 68 subjects collected under 21 lighting and 13 pose conditions. Figure 6.2 shows some example faces of a subject collected under 9 different lighting conditions and 9 pose variations in CMU-PIE face database.

In the experiment on this database, all facial images within five lighting conditions are selected for training and the remaining are used for testing. The face matching is proceeded with the random projection subspace varying from 30% to 90% of the size of the given tensors. The experimental results are shown in Figure 6.3. According to these results, the face matching rate is rather robust with different sizes of the random project subspace. It doesn't change much when the random projection subspace ranges from 50% to 90% of the sizes of the given tensors.

6.1.3 Comparison on CMU-PIE Database

In this experiment, our CSMA method is compared to other state-of-the-art decomposition methods, i.e. PCA, LDA, LPP, MPCA and SPMA, on the challenging CMU-MPIE face database that has been described in section 6.1.2. Our experiment also uses all facial images within five lighting conditions for training and the remaining for testing. This database setup is used to evaluate for all methods mentioned above. The Receiver Operating Char-

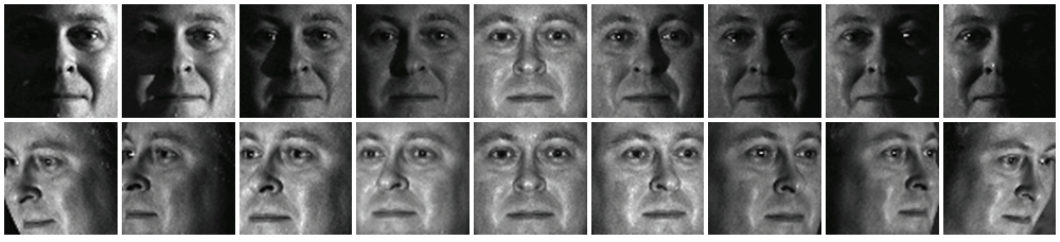


Figure 6.2: Examples on CMU-PIE Database with 9 lighting conditions (first row) and 9 pose variations (second row).

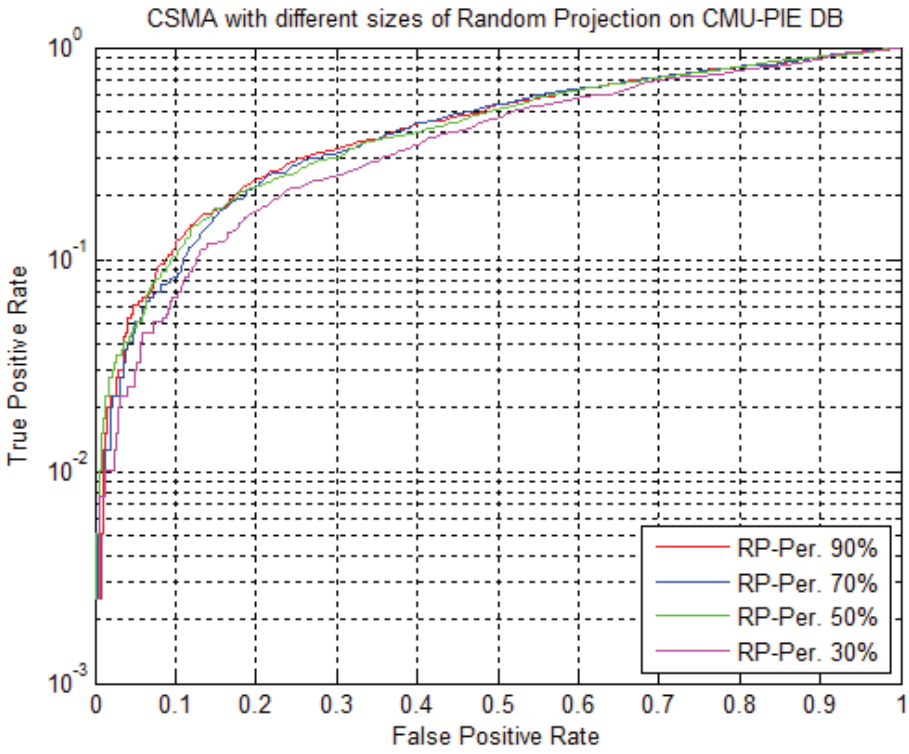


Figure 6.3: CSMA Face Matching on CMU-PIE DB with different sizes of Random Projection subspaces.

acteristic (ROC) curves of the face matching with the cosine distance are computed for all methods and the experimental results are shown in Figure 6.4. According to these results, our proposed method achieves better face matching rate compared to the other methods.

6.1.4 Comparison on Extended YALE-B Database

The Extended YALE-B database [44] has 38 subjects collected under 9 pose and 64 illumination conditions. Some examples of faces changing in pose and illumination are shown in Figure 1.3. Similar to the experiment in section 6.1.3, all facial images in 10 lighting conditions are used for training. The rest that include all facial images in the remaining 54 lighting conditions are used for testing. The Receiver Operating Characteristic curves of the face matching with the cosine distance are also used to evaluate all the methods on this database. The experimental results are shown in Figure 6.4. According to this results,

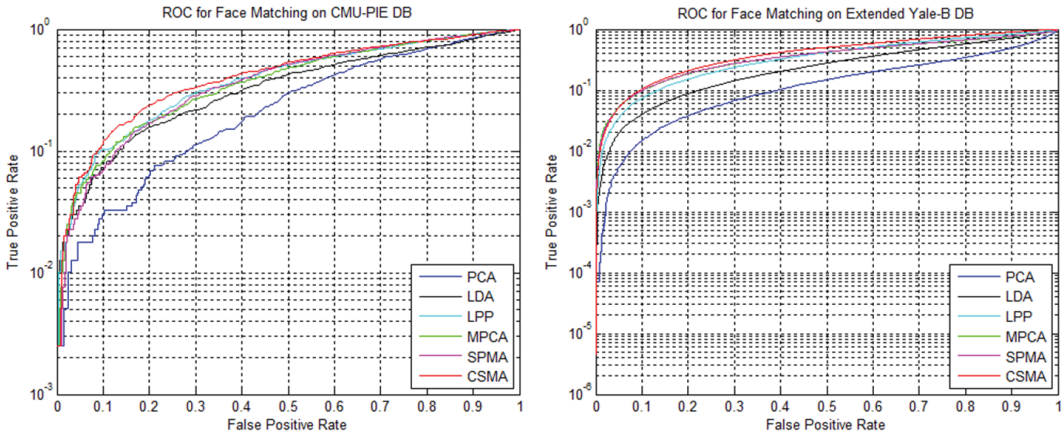


Figure 6.4: Comparison between CSMA and the other subspace decomposition methods on CMU-MPIE DB (left) and Extended Yale-B DB (right).

our proposed method achieves better face matching rate than the other methods, e.g. PCA, LDA, LPP, MPCA and SPMA.

6.2 Background Substraction via SVD-ℓ_1

In this section, we demonstrate that our proposed SVD-ℓ_1 decomposition method presented in chapter 4 can be also applied for the background substraction problem. In this problem, given an off-line video with a fixed camera recording moving objects, e.g. people, cars, animals, etc., we need to distinguish these moving objects against the background so that they can be used as inputs for further processing steps, such as object recognition, activity recognition, tracking, etc.

In our experiments, the method is experimented on a video collected by Elgammal et al. [37, 38]. There are 30 frames with the resolution of 120×160 in three RGB channels recorded in the video. All these frames are setup as column vectors where moving objects, i.e. rain, cars and lighting, can be considered as outliers. Meanwhile the background information in every frame is considered as actual input values. The background and foreground in every frame are then classified using our proposed SVD-ℓ_1 method. Figure 6.5 show the background-foreground classification results. In that figure, the images in the first row are

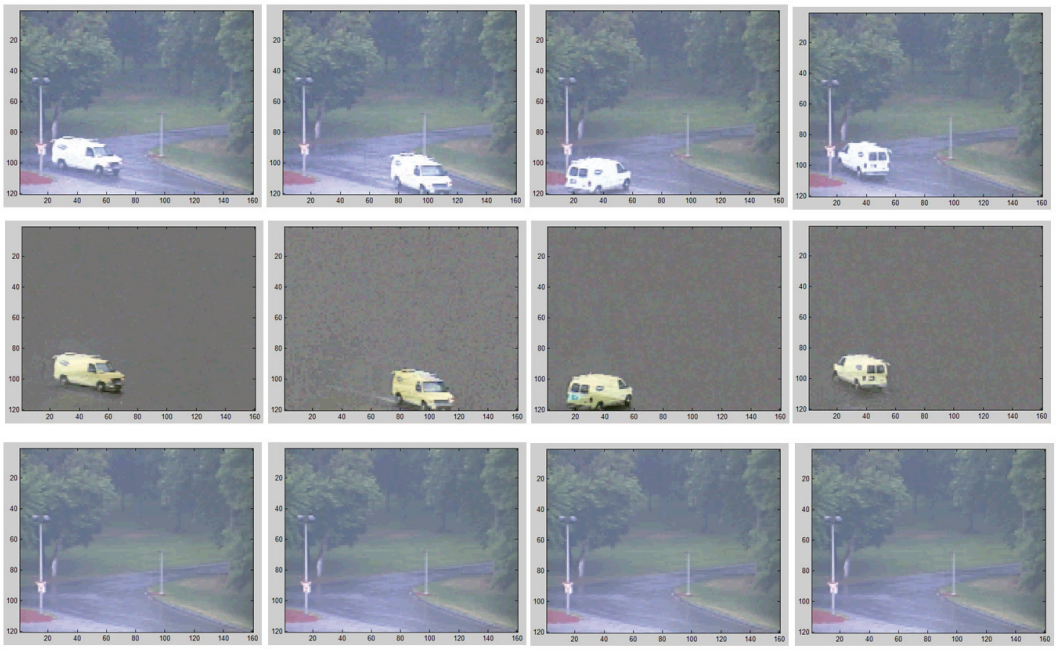


Figure 6.5: An example of background substraction in videos. The images in the first row are from the original video. The corresponding images in the second row are the moving objects extracted from the video. The images in the last row are the background computed using SVD-ℓ_1 on the input video.

some frames recorded in the original video. Meanwhile the images in the second row are the detected foreground (or car) and the images in the third row are the detected background using our proposed method. Our method has ability to adapt the lighting and noise (rain) variations in the video.

6.3 Image Inpainting

Our proposed CSMA method can also be employed efficiently in the inpainting application. In this problem, the input tensor $\mathcal{X}$ is now a degraded image. The method aims to generate the clean image represented by tensor $\tilde{\mathcal{X}}$ without any outliers or noise, e.g. $\mathcal{X} = \tilde{\mathcal{X}} + \mathcal{N}$, where $\mathcal{N}$ depicts outliers or noise. In these experiments, the λ value that controls the trade-off between the low-rank regularization term and the sparsity in Eqn. (4.11) is set to 10^{-3}. The maximum rank of the matrix flattened from the tensor $\mathcal{X}$ will be employed. Since

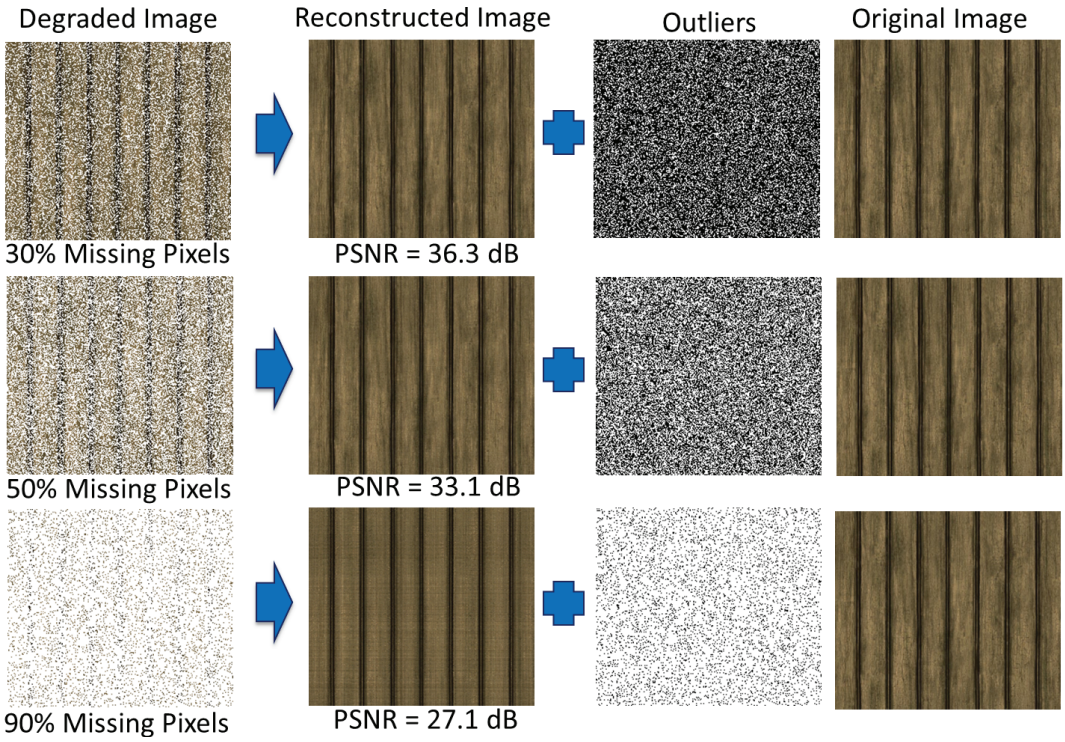


Figure 6.6: An example of CSMA in the inpainting problem with different percentages of missing pixels in an color image of size 250×219 pixels (the first column). The reconstruction results (the second column) show that CSMA can restore a degraded image containing 90% missing values (the third row) with a high accuracy reconstruction (PSNR = 27.1 dB).

the quality of restoration results is the top priority in this application, random projection is therefore set as equal to the size of input images. We also use Peak Signal-to-Noise Ratio as the figure of merit to measured the quality of the restored images in these experiments.

Our proposed CSMA method has ability to restore degraded images containing a large number of missing values. In the first experiment of this section, our tensor-based reconstruction algorithm is evaluated with different percentages of missing values, i.e. 30%, 50% and 90%, of the total number of pixels in the input image. The method is able to reconstruct the degraded images with high accuracy, e.g. high PSNR values, as shown in Fig. 6.6. When the percentages of missing values are up to 90% of the pixels in the input image, the degraded image loses most important information and structure as shown in the first figure of the third row in Fig. 6.6. Human eyes cannot recognize the real pattern behind this degraded image. However, our method is still able to restore it with a high accuracy

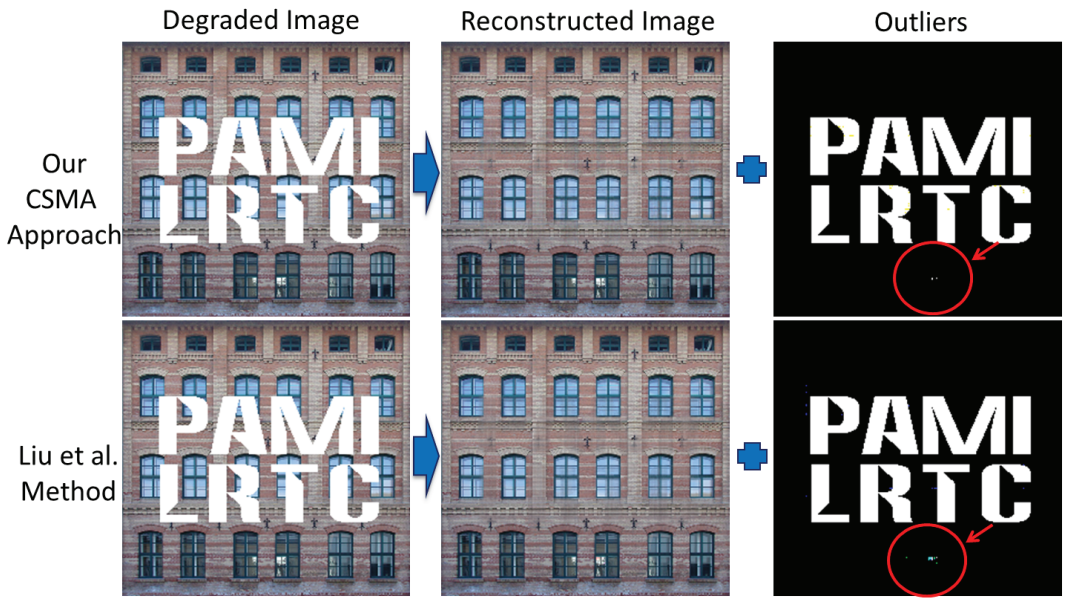


Figure 6.7: The comparison between CSMA and Liu et al. method [66] in the inpainting problem. The red circle shows CSMA gives less reconstruction errors than [66] does in this example. (The comparison of reconstruction errors can be seen clearer when zooming 300%)

rate (PSNR = 27.1 dB) as shown in the second figure of the third row in Fig. 6.6.

In the second experiment, our tensor-based reconstruction algorithm is compared with the other tensor decomposition methods, i.e. Low Rank Tensor Completion (LRTC), Simple LRTC (SiLRTC), SiLRTC without relaxation, Fast LRTC (FaLRTC), FaLRTC without relaxation, presented by Liu et al. [65, 66] in the inpainting application. The experiments are run on a color image of size 250×219 pixels as the image shown in the fourth column in Fig. 6.6. The above-mentioned tensor-based inpainting methods are also evaluated with different percentages of missing values, i.e. 30%, 50% and 90%, of the total number of pixels in the input image. In each case, the PSNR reconstruction results (PSNR) and the computation times (seconds) are computed. Table 6.3 shows the comparison of the experimental results. As the results in this table show the LRTC algorithm consumes less computation time, but their reconstruction accuracy is not as good as the others. The reconstruction result of our method is better than two methods, i.e. LRTC and SiLRTC without relaxation, and is comparable with the other methods, i.e. FaLRTC, FaLRTC and SiLRTC.

Table 6.3: Image Inpainting Comparison between our CSMA method and Liu et al. method [66].

	30% missing		50% missing		90% missing	
Methods	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)	PSNR (dB)	Time (s)
LRTC	32.1	25.4	12.4	15.6	9.7	21.6
FaLRTC	36.3	323.6	33.4	372.4	27.2	502.7
FaLRTC No relaxation	36.5	218.4	33.4	389.6	27.2	745.8
SiLRTC	36.3	359.9	32.5	360.8	22.4	476.9
SiLRTC No relaxation	34.4	280.6	33.4	286.6	9.8	10.5
CSMA	**36.3**	**108.9**	**33.1**	**148**	**27.1**	**397.2**

However, our method requires the lowest computation time among all these methods in all experiments. As shown in Fig. 6.7, our method is able to reconstruct degraded images at least as comparable to the FaLRTC algorithm in [66]. All experiments in this section are processed on a CPU with Intel Core i7, 2.93 GHz and 8 GB of RAM.

6.4 SCFA Experiments

In this section, our proposed Sparse Class-dependent Feature Analysis approach is evaluated on two challenging face databases, e.g. the Extended Yale-B and AR databases. Finally, to show the robustness of the SCFA for any application, results are reported using the Caltech101 dataset as well.

6.4.1 Experiments on Extended YaleB Database

As discussed in section 6.1.4, there are 38 subjects collected under 9 pose and 64 illumination conditions in the Extended Yale-B database. We follow the same evaluation protocol presented in [53, 126], where all frontal images with illumination variation are selected. There are 32 images per subject selected for training and the rest are used for testing. Each facial image is projected onto a 504 dimensional random projection subspace (as

Table 6.4: Classification results on the Extended Yale-B database with the same database selection as in [53]. The first column shows the name of the methods, the second column shows the classification results. The third column shows the number of samples per subject used in dictionaries.

Methods	Acc. (%)	Notice
K-SVD [5]	93.1	15 samples
D-KSVD [126]	94.1	15 samples
SRC [115]	83.3	15 samples
SRC [115]	97.0	32 samples
LC-KSVD2 [53]	95.0	15 samples
LC-KSVD2 [53]	96.7	32 samples
KCFA [93, 117]	88.0	32 samples
SCFA (Our method)	**95.5**	**15 samples**
SCFA (Our method)	**97.5**	**32 samples**

Table 6.5: Classification results on the Extended Yale-B database. In the second column, the results are presented using Mean and Standard Deviation (SD) within 20 times.

Methods	Acc. (Mean$\pm$SD)	Notice
K-SVD [5]	94.83 ± 1.48 %	32 samples
SRC [115]	98.34 ± 0.63 %	32 samples
LC-KSVD1 [53]	94.83 ± 0.95 %	32 samples
LC-KSVD2 [53]	96.41 ± 0.96 %	32 samples
KCFA [93, 117]	87.49 ± 3.8 %	32 samples
SCFA (Our method)	$\mathbf{96.31 \pm 0.52}$ %	**15 samples**
SCFA (Our method)	$\mathbf{98.64 \pm 0.48}$ %	**32 samples**

in [53]). Our proposed approach is compared to K-SVD, D-KSVD, LC-KSVD, sparse representation-based classification (SRC) [115] and the classical KCFA. The performance of our method is evaluated using dictionary sizes of 15 and 32.

The first set of experiments, shown in Table 6.4 is performed in the same manner as that presented in [53]. The data set is partitioned randomly into a training set and testing set and the classification experiment was run once to determine accuracies in Table 6.4. Here, SCFA achieves the best performance amongst the methods. Compared to the classical KCFA, our accuracy increases by about 10%.

In Table 6.5, a better experimental setup is used where the random selection of the training and testing sets as well as the classification experiment is repeated 20 times. For

Table 6.6: Classification results on the AR database.

Methods	Acc. (%)	Notice
K-SVD [5]	91.0	20 samples
SRC [115]	88.17	5 samples
SRC [115]	99.0	20 samples
LC-KSVD1 [53]	89.83	5 samples
LC-KSVD2 [53]	90.67	20 samples
KCFA [93, 117]	86.33	20 samples
SCFA (Our method)	**97.5**	**5 samples**
SCFA (Our method)	**99.17**	**20 samples**

each run, all the discussed methods are evaluated. Finally, the mean and the standard deviation (SD) of the 20 results are computed for every method. These values are reported in Table 6.5. Here too, we see that SCFA achieves the highest accuracy with a mean of 98.64%. It is 10% better than the result from KCFA and 2% better than that from LC-KSVD2. In addition our method reports the least standard deviation (0.48) as well as less computation time than SRC (see Table 6.7), thus showing its versatility and practicality. Figure 6.8 shows some sample faces recognized with 100% classification rate.

6.4.2 Experiments on AR Database

The AR database [75] has 4,000 facial images collected from 126 subjects as shown in Figure 6.8. Each subject has 26 facial images with different facial expressions, illumination conditions and occlusions. We follow the evaluation protocol presented in [53, 126]. There are 2,600 facial images selected from 50 males and 50 females. In the same manner as in [53] for dimensionality reduction, each facial image is projected onto a 504 dimen-

Table 6.7: Computation time for recognizing a test face image on the Extended Yale-B database on a CPU with Intel Core i7, 2.93 GHz and 8 GB of RAM.

Methods	Avg. Time (ms)	Notice
SRC [115]	82.5	15 samples
SRC [115]	91.1	32 samples
SCFA (Our method)	**2.6**	**15 samples**
SCFA (Our method)	**4.9**	**32 samples**

Table 6.8: Computation time for recognizing a test face image on the AR database.

Methods	Avg. Time (ms)	Notice
SRC [115]	87.6	5 samples
SRC [115]	127	10 samples
SCFA (Our method)	**2.8**	**5 samples**
SCFA (Our method)	**4.3**	**10 samples**

Figure 6.8: Example training and testing images in Extended YaleB (first two rows) and AR databases (the third row) classified with 100% accuracy.

sional random projection subspace. The performance of our method is evaluated using two different dictionary sizes of 5 and 20.

Our SCFA approach is also compared to the methods described in section 6.4.1. The results are shown in Table 6.6. Our method achieves the best accuracy rates and we see that it is 10% better than the classical KCFA approach. We also compare our computation time with that of SRC (which also reports high accuracies) in Table 6.8 and we see the gain in time when using our proposed method.

6.4.3 Experiments on Caltech101 Dataset

The Caltech101 database [41] consists of 9,144 images from 101 classes of objects and one class of backgrounds. These objects are collected with large appearance variations, different illumination and materials as shown in Figure 6.9. Due to this, classical KCFA is unable to achieve the high classification on this database due to the huge variations of objects. We follow the evaluation protocol presented in [53], with 30 images per subject

Table 6.9: Classification results with different number of training images per subject on the Caltech101 dataset

Num. of Train. Samples	5	10	15	20	30
Malik [125]	46.6	55.8	59.1	62.0	66.2
Griffin [46]	44.2	54.5	59.0	63.6	67.6
Wang [112]	51.2	59.8	65.4	67.7	73.4
SRC [115]	54.6	62.5	67.3	69.7	72.5
K-SVD [5]	70.2	70.5	70.7	71.7	73.9
LC-KSVD1 [53]	61.5	67.6	69.1	70.2	73.8
LC-KSVD2 [53]	61.6	68.18	70.4	73.2	74.0
KCFA [93, 117]	61.0	62.8	65.5	68.6	70.8
SCFA (Our method)	**72.9**	**73.0**	**73.1**	**73.2**	**74.9**

Figure 6.9: Example images in Caltech database.

for training and the rest for testing. Here, in addition to methods compared with so far, we compare our proposed approach with other state-of-the-art methods used with this database, e.g. [46, 112, 125]. Different numbers of training images per subject are used, e.g. 5, 10, 15, 20 and 30, in these experiments. Table 6.9 tabulates our relevant results. Here too we see the superiority of the proposed approach compared to the state of the art.

Chapter 7

Conclusion

This thesis has built a bridge to fill the gap between the tensor decomposition research topics to analyze relationships among factors from given tensors and the Compressed Sensing, one of the hottest research topics nowadays. In first few chapters, we have briefly reviewed the terminologies as well as the fundamental backgrounds presented in multifactor analysis and Compressed Sensing problems. In chapter 4, we have then introduced a novel Compressed Submanifold Multifactor Analysis approach in order to analyze any given multifactor data with missing values and outliers. Compared to the state-of-the-art Multilinear PCA method that averages the shapes of each factor and therefore loses the original structures, our approach has the ability to keep geometrical structures of the input factors robustly. More importantly, our proposed method also provides the capability efficiently to handle missing values and detect outliers and noisy values from given tensors. These achievements were obtained thank to the power of the ℓ_1-based Higher-order Singular Value Decomposition method proposed in section 4.2.1 in this thesis. The proposed CSMA is then evaluated and achieved the outperformed recognition accuracy compared to the standard multifactor decomposition methods, i.e. Multilinear PCA, Submanifold Multifactor Analysis and others, on two challenging databases, i.e. CMU-MPIE and Extended YALE-B. The face recognition accuracy obtained by these experiments shows the robust-

ness of our CSMA method compared to the others. These results are due to the fact that our CSMA method can deal with missing values, detect outliers and ignore noisy values. Additionally, the parameters in our method can preserve the structure of factor-dependent submanifolds in the space generated from input data which is impossible in Multilinear PCA due to its parameter averaging process.

In addition, in the second part of this thesis, we have also presented a novel dictionary based nonlinear classification model, named Sparse Class-dependent Feature Analysis. The method benefits from the use of both sparse representation in a dictionary and class specific optimal filters. The performance improvement is due to our powerful non-linear classification tool optimized in tandem with a highly flexible feature representation. In summary, our method outperforms the state of the art in face recognition on Extended Yale-B and AR databases and object recognition on Caltech101 database. Hence, the proposed method can depict its wide applicability to solve general computer vision and pattern recognition problems.

List of author's publications related to this thesis work:

1. **K. Luu**, M. Savvides, T. D. Bui and C. Y. Suen. Compressed Submanifold Multifactor Analysis with Adaptive Factor Structures, International Conference on Pattern Recognition (ICPR). Japan, Nov. 2012.
2. **K. Luu**, M. Savvides, T. D. Bui and C. Y. Suen. Compressed Submanifold Multifactor Analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI), Jan. 2013 (under review, round 2).
3. **K. Luu**, S. Venugopalan, H. N. Le, M. Savvides and T. D. Bui. Sparse Class Dependent Feature Analysis. IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI), Aug. 2013 (under review).
4. H. N. Le, **K. Luu** and M. Savvides. SparCLeS: Dynamic L1 Sparse Classifiers with Level Sets for Robust Beard/Moustache Detection and Segmentation. IEEE Transaction on Image Processing (TIP), Volume 22, Number 8, Pages 3097-3107, Aug. 2013.
5. H. N. Le, K. Seshadri, **K. Luu** and M. Savvides. Facial Aging and Asymmetry Decomposition Based Approaches to Identification of Twins. IEEE Transaction on Image Processing (TIP), Jan. 2013 (under review, round 3).
6. F. J. Xu, **K. Luu** and M. Savvides. Spartans: Single-sample Periocular-based Alignment-robust Recognition Technique Applied to Non-frontal Scenarios. IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI), Nov. 2013. (Under review)
7. H. N. Le, **K. Luu**, K. Seshadri and M. Savvides. A Facial Aging Approach to Identification of Identical Twins. IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington DC, Sep. 2012.
8. Y. Xie, **K. Luu** and M. Savvides. A Robust Approach to Facial Ethnicity Classification on Large Scale Face Databases. IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), Washington DC, Sep. 2012.

9. **K. Luu**, Yiting Xie, and M. Savvides. Class-dependent L1-min Feature Analysis for Robust Ethnicity Classification on Large Scale Face Databases. Pattern Recognition Letters, Jan. 2014 (under review).

10. **K. Luu**, T. D. Bui, C. Y. Suen, K. Ricanek. Spectral Regression based Age Determination. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), San Francisco, 2010.

11. **K. Luu**, T. D. Bui, and C. Y. Suen. Kernel Spectral Regression of Perceived Age from Hybrid Facial Features. IEEE Conference on Automatic Face and Gesture Recognition (FGR), Santa Barbara, March 2011.

Bibliography

[1] Computing in Science and Engineering. http://galton.uchicago.edu/~lekheng/courses/309f11/top10. Accessed: 2013-07-10.

[2] FaceIt SDK, Morphotrust. http://www.morphotrust.com/. Accessed: 2013-07-10.

[3] NSF Workshop, Future Directions in Tensor-based Computation and Modeling. http://www.cs.cornell.edu/cv/TenWork/Home.htm. Accessed: 2013-07-10.

[4] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2D and 3D Face Recognition: A Survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007.

[5] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov. 2006.

[6] R. Baraniuk. Compressive Sensing. *IEEE Signal Processing Magazine*, 24(4):118–120, 2007.

[7] A. Bartoli, D. Pizarro, and M. Loog. Stratified Generalized Procrustes Analysis. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 1–10, 2010.

[8] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):711–720, 1997.

[9] M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15:1373–1396, 2003.

[10] S. Bendapudi, K. Luu, and M. Savvides. Hallucinating faces in the dark. In *Proceedings of IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 311–318, 2012.

[11] F. Bergeaud and S. Mallat. Matching Pursuit of Images. In *Proceedings of International Conference on Image Processing*, pages 53–56, 1995.

[12] E. Bingham and H. Mannila. Random Projection in Dimensionality Reduction: Applications to Image and Text Data. In *Knowledge Discovery and Data Mining*, pages 245–250. ACM Press, 2001.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. 3(1):1–122, 2011.

[14] O. Bryt and M. Elad. Compression of Facial Images using the K-SVD Algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008.

[15] J. F. Cai, E. J. Candes, and Z. Shen. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[16] E. Candes. Compressive Sampling. In *Proceedings of the International Congress of Math.*, pages 1–20, 2006.

[17] E. Candes. The Restricted Isometry Property and Its Implications for Compressed Sensing. *Compte Rendus de l'Academie des Sciences, Paris, Serie I*, 346:589–592, 2008.

[18] E. Candes, J. Romberg, and T. Tao. Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[19] E. Candes, J. Romberg, and T. Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[20] E. Candes and T. Tao. Near Optimal Signal Recovery from Random Projections: Universal Encoding Strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.

[21] E. Candes and M. B. Walkin. An Introduction to Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2005.

[22] J. D. Carroll and J. J. Chang. Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of "Eckart-Young" Decomposition. *Psychometrika*, 35:283–319, 1970.

[23] R. B. Cattell. Parallel Proportional Profiles and Other Principles for Determining the Choice of Factors by Rotation. *Psychometrika*, 9:267–283, 1944.

[24] S. S. Chen, D. Donoho, and A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[25] T. F. Cootes, C. J. Taylor, and A. Lanitis. Active Shape Models : Evaluation of a Multi-resolution Method for Improving Image Search. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 327–336, 1994.

[26] S. Dasgupta and A. Gupta. An elementary proof of the Johnson Lindenstrauss lemma. Technical report, TR-99-006, U.C.Berkeley, 1999.

[27] H. A. Van der Vorst. Krylov Subspace Iteration. *Computing in Science and Engineering*, 2(1):32–37, 2000.

[28] J. Dongarra and F. Sullivan. Introduction to The Top 10 Algorithms. *Computing in Science and Engineering*, 2(1):22–23, 2000.

[29] D. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*,

52(4):1289–1306, 2006.

[30] D. Donoho and Y. Tsaig. Fast Solution of l1-norm Minimization Problems When the Solution May Be Sparse. Technical report, Department of Statistics, Stanford University, 2006.

[31] D. L. Donoho and I. M. Johnstone. Adapting to Unknown Smoothness via Wavelet Shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.

[32] M. Duarte, M. Davenport, T. Sun K. Kelly D. Takhar, J. Laska, and R. Baraniuk. Single-pixel Imaging via Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

[33] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators. *Mathematical Programming*, 55:293–318, 1992.

[34] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

[35] M. Elad. *Sparse and Redundant Representations*. Springer, 2010.

[36] M. Elad and M. Aharon. Image Denoising via Sparse and Redundant Representations over Learned Dictionaries. *IEEE Transactions on Image Processing (TIP)*, 15(12):3736–3745, 2006.

[37] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Non-parametric Model for Background Subtraction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 751–767, 2000.

[38] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and Foreground Modeling using Non-parametric Kernel Density Estimation for Visual Surveillance. In *Proceedings of the IEEE*, pages 1151–1163, 2002.

[39] A. Eriksson and A. Van den Hengel. Efficient Computation of Robust Low-rank

Matrix Approximations in the Presence of Missing Data using the L1 Norm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 771–778, 2010.

[40] C. Faloutsos, T. G. Kolda, and J. Sun. Mining Large Time-evolving Data using Matrix and Tensor Tools. In *Proceedings of International Conference on Machine Learning (ICML), Tutorial*, 2007.

[41] L. FeiFei, R. Fergus, and P. Perona. Learning Generative Visual Models from Few Training Samples: An Incremental Bayesian Appoach Tested on 101 Object Categories. *Journal of Computer Vision and Image Understanding (CVIU)*, 106(1):59–70, 2007.

[42] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1:586–597, 2007.

[43] W. Freeman and J. Tenenbaum. Learing Bilinear Models for Two-factor Problems in Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 554–560, 1997.

[44] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From Few to Many: Illumination Cone Models for Face recognition under Variable Lighting and Pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):643–660, 2001.

[45] J. C. Gower and G. B. Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004.

[46] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. In *CIT Technical Report 7694*, 2007.

[47] R. Gross, L. Sweeney, and F. de la T.and S. Baker. Semi-Supervised Learning of Multi-Factor Models for Face De-Identification. In *Proceedings of IEEE Interna-*

tional Conference on Computer Vision (ICCV), pages 1–8.

[48] R. A. Harshman. Foundations of the PARAFAC Procedure: Models and Conditions for an Explanatory Multi-modal Factor Analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[49] J. Hastad. Tensor Rank is NP-Complete. *Journal of Algorithms*, 11(4):644–654, 1990.

[50] X. He, D. Cai, S. Yan, and H. J. Zhang. Neighborhood Preserving Embedding. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV)*, pages 1208–1213, 2005.

[51] X. He and P. Niyogi. Locality Preserving Projections. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2003.

[52] F. L. Hitchcock. The Expression of a Tensor or a Polyadic as a Sum of Products. *Journal of Mathematical Physics*, 6:164–189, 1927.

[53] Z. Jiang, Z. Lin, and L. S. Davis. Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1697–1704, 2011.

[54] W. B. Johnson and J. Lindenstrauss. Extensions of Lipshitz mapping into Hilbert space. *Contemporary Math.*, 1(26):189–206, 1984.

[55] I. T. Jolliffe. *Principal Component Analysis*. Second Edition, Springer, 2002.

[56] A. Kapteyn, H. Neudecker, and T. Wansbeek. An Approach to N-mode Components Analysis. *Psychometrika*, 51:269–275, 1986.

[57] Qifa Ke and Takeo Kanade. Robust L1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 592–599, 2005.

[58] T. Kolda and B. Bader. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009.

[59] K. Kreutz-Delgado, B. D. Rao, and K. Engan. Novel Algorithms for Learning Overcomplete Dictionaries. In *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers*, volume 2, pages 971–975, 1999.

[60] P. M. Kroonenberg and J. De Leeuw. Principal Component Analysis of Three-mode Data by Means of Alternating Least Squares Algorithms. *Psychometrika*, 45:69–97, 1980.

[61] N. Kwak. Principal Component Analysis based on L1-Norm Maximization. *IEEE Transaction on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(9):1672–1680, 2008.

[62] L. D. Lathauwer, B. D. Moor, and J. Vandewalle. A Multilinear Singular Value Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21:1253–1278, 2000.

[63] H. Soo Lee and D. Kim. Tensor-based AAM with Continuous Variation Estimation: Application to Variation-robust Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(6):1102–1116, March 2009.

[64] J. Liu. Generalized Low Rank Approximations of Matrices Revisited. *IEEE Transaction on Neural Networks*, 21(4):621–632, 2010.

[65] J. Liu, P. Musialski, P. Wonka, and Y. Jieping. Tensor completion for estimating missing values in visual data. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2114–2121, 2009.

[66] J. Liu, P. Musialski, P. Wonka, and Y. Jieping. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[67] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. A Survey of Multilinear Sub-

space Learning for Tensor Data. *Journal of Pattern Recognition*, 44(7):1540–1551, 2011.

[68] M. Lustig, D. Donoho, and J. Pauly. Rapid MR Imaging with Compressed Sensing and Randomly Under-sampled 3DFT Trajectories. In *Proceedings of Annual Meeting of ISMRM*, 2006.

[69] M. Lustig, J. Lee, D. Donoho, and J. Pauly. Faster Imaging with Randomly Perturbed, Under-sampled Spirals and l1 Reconstruction. In *Proceedings of Annual Meeting of ISMRM*, 2005.

[70] K. Luu, M. Savvides, T. D. Bui, and C. Y. Suen. Compressed Submanifold Multifactor Analysis with Adaptive Factor Structures. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 2715–2718, 2012.

[71] A. Mahalanobis, B. V. K. V. Kumar, and D. Casasent. Minimum Average Correlation Energy Filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 153–153, 2005.

[72] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative Learned Dictionaries for Local Image Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[73] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative Sparse Image Models for Class-Specific Edge Detection and Image Interpretation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 43–56, 2008.

[74] S. G. Mallat and Z. Zhang. Matching Pursuits with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[75] A.M. Martinez and R. Benavente. The AR Face Database. CVC Technical Report No. 24, June 1998.

[76] M. R. Osborne, B. Presnell, and B. A. Turlach. A New Approach to Variable Se-

lection in Least Squares Problems. *IMA Journal of Numerical Analysis*, 3:389–403, 2000.

[77] Y. Pang, X. Li, and Y. Yuan. Robust Tensor Analysis with L1-Norm. *IEEE Transaction on Circuits and Systems for Video Tech.*, 20(2):172–178, 2010.

[78] S. Park. *Multifactor Analysis for Face Recognition Based on Factor-Dependent Geometry*. Ph.D Thesis, Carnegie Mellon University, 2011.

[79] S. Park and M. Savvides. An extension of multifactor analysis for face recognition based on submanifold learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2645–2652, 2010.

[80] S. Park and M. Savvides. Multifactor analysis based on factor-dependent geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2817–2824, 2011.

[81] Sung Won Park and Marios Savvides. A multifactor extension of linear discriminant analysis for face recognition under varying pose and illumination. *EURASIP Journal on Advances in Signal Processing*, pages 1156–1166, 2010.

[82] B. N. Parlett. The QR Algorithm. *Computing in Science and Engineering*, 2(1):38–42, 2000.

[83] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *Proceedings of Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993.

[84] D. Pizarro and A. Bartoli. Global Optimization for Optimal Generalized Procrustes Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2409–2415, 2011.

[85] U. Prabhu, J. Heo, and M. Savvides. Unconstrained pose-invariant face recognition using 3d generic elastic models. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, 33(10):1952–1961, 2011.

[86] P. Refregier. Filter Design for Optical Pattern Recognition: Multi-criteria Optimization Approach. *Journal of Optics Letter*, 15:854–856, 1990.

[87] J. Romberg. Compressive Sensing by Random Convolution. *SIAM Journal on Imaging Sciences*, 4(2):1098–1128, 2009.

[88] J. Romberg and M. Wakin. Compressed Sensing: A Tutorial. `http://users.ece.gatech.edu/~justin/ssp2007`, Aug 2007.

[89] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 22:2323–2326, 2000.

[90] A. C. Sankaranarayanan. Compressive Sensing and Sparse Optimization, Special Topics in Signal Processing, Spring 2013. `http://users.ece.cmu.edu/~saswin/files/teaching/Spring13/`. Accessed: 2013-07-10.

[91] L. K. Saul and S. T. Roweis. An Introduction to Locally Linear Embedding. Technical report, AT&T Labs, 2000.

[92] B. Savas and L. Eldn. Krylov-type Methods for Tensor Computations. *Preprint arXiv:1005.0683[math.NA]*, 2011.

[93] M. Savvides, B. V. K. V. Kumar, and P. K. Khosla. "Corefaces" - Robust Shift Invariant PCA Based Correlation Filter for Illumination Tolerant Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 834–841, 2004.

[94] K. Seshadri and M. Savvides. Robust Modified Active Shape Model for Automatic Facial Landmark Annotation of Frontal Faces. In *Proceedings of the 3rd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 319–326, 2009.

[95] K. Seshadri and M. Savvides. An Analysis of the Sensitivity of Active Shape Models

to Initialization when Applied to Automatic Facial Landmarking. *IEEE Transactions on Information Forensics and Security (TIFS)*, 7(4):1255–1269, 2012.

[96] T. Sim, S. Baker, and M. Bsat. The CMU Pose, Illumination, and Expression Database. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(12):1615–1618, 2003.

[97] D. Smeets, P. Claes, J. Hermans, D. Vandermeulen, and P. Suetens. A comparative study of 3-d face recognition under expression variations. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 42(5):710–727, 2012.

[98] G. W. Stewart. The Decompositional Approach to Matrix Computation. *Computing in Science and Engineering*, 2(1):50–59, 2000.

[99] D. Strelow. General and Nested Wiberg Minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1584–1591, 2012.

[100] J. Sun, D. Tao, S. Papadimitriou, P. Yu, and C. Faloutsos. Incremental Tensor Analysis: Theory and Applications. *ACM Transaction on Knowledge Discovery Data*, 2(3):1–37, 2008.

[101] T. Tao. Compressed Sensing Revisited. In *Mahler Lecture Series*, 2008.

[102] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science 290*, 22:2319–2323, 2000.

[103] L. R. Tucker. Implications of Factor Analysis of Three-way Matrices for Measurement of Change. *in Problems in Measuring Change, C. W. Harris, ed., University of Wisconsin Press*, pages 122–137, 1963.

[104] L. R. Tucker. Some Mathematical Notes on Three-mode Factor Analysis. *Psychometrika*, 31:279–331, 1966.

[105] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991.

[106] L. J. P. van der Maaten and G. E. Hinton. Visualizing high-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[107] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality Reduction: A Comparative Review. Technical report, TiCC-TR-2009-005, Tilburg University, 2009.

[108] M. Vasilescu and D. Terzopoulos. Multilinear Image Analysis for Facial Recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 511–514, 2002.

[109] M. Vasilescu and D. Terzopoulos. Multilinear Independent Components Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 547–553, 2005.

[110] M. A. O. Vasilescu and D. Terzopoulos. Multilinear Analysis of Image Ensembles: Tensor Faces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–460, 2002.

[111] S. S. Vempala. *The Random Projection Method*. American Math. Society, 2004.

[112] J. Wang, J. Yang, K. Yu, F. Lv amd T. Huang, and Y. Gong. Locality-constrained Linear Coding for Image Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367, 2010.

[113] J. Wright, Y. Ma, J. Mairal, G. Spairo, T. Huang, and S. Yan. Sparse Representation for Computer Vision and Pattern Recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[114] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(2).

[115] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(2):210–227, Feb. 2009.

[116] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse Reconstruction by Separable Approximation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3373–3376, 2008.

[117] C. Xie, M. Savvides, and B. V. K. V. Kumar. Redundant Class-Dependence Feature Analysis Based on Correlation Filters Using FRGC2.0 Data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 153–153, 2005.

[118] Y. Xie, K. Luu, and M. Savvides. A Robust Approach to Facial Ethnicity Classification on Large Scale Face Databases. In *Proceedings of the IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 143–149, 2012.

[119] J. Xu, K. Luu, M. Savvides, T. D. Bui, and C. Y. Suen. Investigating age invariant face recognition based on periocular biometrics. In *Proceedings of IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–7, 2011.

[120] J. Yang, K. Yu, and T. Huang. Supervised Translation-invariant Sparse Coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3517–3524, 2010.

[121] J. Yang, D. Zhang, J. Yang, , and B. Niu. Globally Maximizing, Locally Minimizing: Unsupervised Discriminant Projection with Applications to Face and Palm Biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29:650–664, 2007.

[122] J. Yang and Y. Zhang. Alternating Direction Algorithms for l1-problems in Compressive Sensing. Technical report, 2009.

[123] J. Ye. Generalized Low Rank Approximations of Matrices. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.

[124] X. Yuan and J. Yang. Sparse and Low-rank Matrix Decomposition via Alternating Direction M]ethods, institution = Department of Math., Hong Kong Baptist University, year = 2009. Technical report.

[125] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2126–2136, 2006.

[126] Q. Zhang and B. Li. Discriminative K-SVD for Dictionary Learning in Face Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2691–2698, 2010.

[127] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips. Face Recognition: A Literature Survey. *ACM Computer Survey*, 35(4):399–458, 2003.

[128] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi. Practical Low-Rank Matrix Approximation under Robust L1-Norm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1410–1417, 2012.