ROBUST DATA REPRESENTATIONS FOR VISUAL LEARNING

A Dissertation Presented
by

SHENG LI

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in

Computer Engineering

Northeastern University
Boston, Massachusetts

April 2017
To my wife.
Contents

List of Figures ................................................................. vi
List of Tables ................................................................. ix
Acknowledgments .............................................................. xi
Abstract of the Dissertation ................................................. xii

1 Introduction ..................................................................... 1
  1.1 Background .............................................................. 1
  1.2 Related Work ........................................................... 3
    1.2.1 Subspace Learning ................................................. 3
    1.2.2 Subspace Clustering .............................................. 4
    1.2.3 Low-Rank Modeling .............................................. 4
    1.2.4 Multi-View Learning ............................................ 6
    1.2.5 Self-Taught Learning ............................................ 6
  1.3 Dissertation Organization ............................................. 7

2 Robust Graph Construction ............................................. 9
  2.1 Background and Motivation ........................................... 9
  2.2 Existing Graph Construction Methods ............................... 12
    2.2.1 Unbalanced Graphs and Balanced Graph .................. 12
    2.2.2 Sparse Representation based Graphs ....................... 13
    2.2.3 Low-Rank Learning based Graphs ............................ 13
  2.3 Low-Rank Coding based Unbalanced Graph Construction ........ 14
    2.3.1 Motivation ....................................................... 14
    2.3.2 Problem Formulation .......................................... 16
    2.3.3 Complexity Analysis .......................................... 19
    2.3.4 Discussions .................................................... 19
  2.4 Low-Rank Coding based Balanced Graph Construction .......... 19
    2.4.1 Motivation and Formulation ................................. 20
    2.4.2 Optimization .................................................... 21
  2.5 Learning with Graphs ................................................ 22
    2.5.1 Graph based Clustering ....................................... 22
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Experiments</td>
<td>88</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Settings</td>
<td>88</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Results on Gesture Dataset</td>
<td>90</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Results on Action Datasets</td>
<td>91</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Discussions</td>
<td>96</td>
</tr>
<tr>
<td>5.4</td>
<td>Summary</td>
<td>98</td>
</tr>
<tr>
<td>6</td>
<td>Robust Dictionary Learning for Knowledge Transfer</td>
<td>99</td>
</tr>
<tr>
<td>6.1</td>
<td>Background and Motivation</td>
<td>99</td>
</tr>
<tr>
<td>6.2</td>
<td>Self-Taught Low-Rank (S-Low) Coding</td>
<td>103</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Motivation</td>
<td>103</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Problem Formulation</td>
<td>103</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Optimization</td>
<td>107</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Algorithm and Discussions</td>
<td>108</td>
</tr>
<tr>
<td>6.3</td>
<td>Learning with S-Low Coding</td>
<td>110</td>
</tr>
<tr>
<td>6.3.1</td>
<td>S-Low Clustering</td>
<td>110</td>
</tr>
<tr>
<td>6.3.2</td>
<td>S-Low Classification</td>
<td>110</td>
</tr>
<tr>
<td>6.4</td>
<td>Experiments</td>
<td>111</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Datasets and Settings</td>
<td>112</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Property Analysis</td>
<td>115</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Clustering Results</td>
<td>117</td>
</tr>
<tr>
<td>6.4.4</td>
<td>Classification Results</td>
<td>118</td>
</tr>
<tr>
<td>6.4.5</td>
<td>Discussions</td>
<td>121</td>
</tr>
<tr>
<td>6.5</td>
<td>Summary</td>
<td>122</td>
</tr>
<tr>
<td>7</td>
<td>Conclusion</td>
<td>123</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>125</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Flowchart of our framework. It contains three parts: similarity metric, graph sparsification and applications. (1) Low-rank representation coefficients are utilized to measure the similarities between all pairs of samples, and a complete graph can then be built, where the bold edges imply larger graph weights. (2) $k$-NN and $b$-matching constraints are employed for graph sparsification. The first two parts are learnt iteratively. (3) The applications based on our graph include clustering, transductive and inductive semi-supervised classification. ............................................. 10

2.2 Similarity measurements in the original space and low-rank code space. .......................... 15

2.3 Convergence curve of our approach on PIE database. ($\rho = 1.2$, $\mu = 10^{-3}$ and $\epsilon = 10^{-5}$) ................................................................. 27

2.4 AC and NMI of our graph under different values of $\lambda_1$ and $\lambda_2$ on Extended YaleB face database. ................................................................. 28

2.5 Accuracy of transductive semi-supervised classification of our models and compared methods versus different percentages of labeled samples on four databases. ............. 30

2.6 Accuracy of inductive semi-supervised classification of our approach and compared methods versus different percentages of labeled samples on four databases. ........... 31

2.7 Visualization of different weighted graphs on PIE face database. (Please enlarge the figure to see details in graphs) ...................................................... 33

3.1 Framework of the proposed approach. We jointly remove noise from data $X$ and learn robust subspace $P$. The corrupted samples are mixed in the original space, but they are well separated in the learned subspace. .............................. 36

3.2 Recognition rates of SRRS with different values of $\lambda_1$ and $\lambda_2$ on COIL dataset. ............ 48

3.3 Recognition rates of our approach and compared subspace methods versus varying feature dimensions on the original ((a)-(d)) and corrupted ((e)-(h)) COIL object database. Note that LDA and our approach obtains at most $c - 1$ features, where $c$ is the number of classes. ................................................................. 49

3.4 Average recognition rates of all compared methods on COIL database with different level of noise. ................................................................. 50

3.5 Properties of our approach on ALOI dataset. (a) Convergence curve ($\rho = 1.3$, $\mu = 0.1$ and $\epsilon = 10^{-8}$). (b) Recognition rates of SRRS with different values of $\eta$. ............. 51

3.6 Sample images in YaleB, FERET and KinFace datasets. ............................................ 53
3.7 Average recognition rates of all compared methods on YaleB database with different level of noise. 54
3.8 Visualization of low-rank representation coefficients of two pairs of samples on FERET database. 56
3.9 Visualization of corrupted images ($X$), recovered images ($AZ$) and noise ($E$) on five object and face databases. 58
3.10 Training time (seconds) with different number of training samples on COIL object database. 58

4.1 Framework of our MDBP approach. The Multimodal Spoken Word dataset contains three data views, including video, audio, and magnetic sensors. MDBP maps multi-view data onto a shared subspace through a pair of view-specific bilinear projections (i.e., $P_v$ and $Q_v$), and incorporates discriminative information by enhancing the between-class separability. $Y_i$ is the compact representation of $i$-th class in the latent space. 62
4.2 Classification accuracy (%) with different dimensions on View-1 of UCI Daily and Sports Activity dataset. 74
4.3 Classification accuracy (%) with different dimensions on View-1 of Multimodal Spoken Word dataset. 76
4.4 Parameter sensitivity of $\lambda_1$ and $\lambda_2$ in our approach on UCI Daily and Sports Activity dataset (View-1). The indexes from 1 to 13 on x/y axis correspond to parameters \{0, $10^{-4}$, $5 \times 10^{-4}$, $10^{-3}$, $5 \times 10^{-3}$, $10^{-2}$, $5 \times 10^{-2}$, 0.1, 0.5, 1, 5, 10, 20\}. 77
4.5 (a) Parameter sensitivity of $s$ and (b) convergence curve of our approach on UCI Daily and Sports Activity dataset. 78

5.1 Framework of the proposed approach. We first extract frame-level features from video sequence, and then learn a non-negative dictionary and the representation codings, with the help of temporal regularization. Multiple temporal segments are finally generated via clustering on the codings. 81
5.2 14 gestures in Keck dataset. 89
5.3 Clustering results on Keck dataset. 14 colors denote 14 different gestures. (Please view the color figure for better visualization) 90
5.4 10 actions in Weizmann dataset. 91
5.5 Clustering results on Weizmann dataset. Different colors denote different actions. (Please view the color figure for better visualization) 92
5.6 Example frames of MAD dataset. 94
5.7 Clustering results on MAD dataset (Subject 1) with different number of actions. (Left: Accuracy. Right: NMI) 95
5.8 Visualization of graphs learned by SSC, LRR, LSR, OSC and our TSC approach on Keck gesture dataset. The red color denotes large graph weights, while the blue color indicates small weights. 95
5.9 Sensitivity analysis on Weizmann dataset. Left: accuracy with different values of parameters; Right: accuracy with different number of clusters. 96
5.10 Clustering results with different dictionary size (Left) and different number of sequential neighbors (Right) on Keck dataset. 97
6.1 Diagram of the self-taught low-rank (S-Low) coding framework. A small target dataset \( X_T \) is usually not sufficient to extract effective features. By utilizing the auxiliary dataset \( X_S \), the proposed S-Low framework learns a shared dictionary \( D \) from two domains, and enforces a low-rank constraint on the coefficient matrix of target domain \( Z_T \) that is considered as new feature representations. Finally, the normalized cut (NCut) algorithm can be utilized for image clustering, and the support vector machine (SVM) can be trained on \( Z_T \) for image classification.

6.2 Sample images in auxiliary domain (above) and target domain (below).

6.3 Convergence property of our approach on Caltech-101 dataset, measured by (a) relative error of \( Z_T \); (b) relative error of \( D \).

6.4 Accuracy of Ours-B approach under different values of \( \lambda_1, \lambda_2 \) and \( \lambda_3 \) on MSRC-v1 dataset.

6.5 Parameter sensitivity of our approach on Caltech-101 dataset: (a) clustering accuracy with different dictionary sizes; (b) NMI with different values of \( K \).

6.6 Classification accuracy of our approach with different sizes of dictionary on MSRC-v1 and Scene-15 datasets.

6.7 (a-b) Confusion matrices of our approach on MSRC-v1 and Scene-15 datasets. (c) Example images from classes with high classification accuracy of Caltech-101.

6.8 Visualization of graphs learnt by (a) Latent LRR, (b) RDSTL and (c) Our approach on MSRC-v1 image dataset. The red color denotes large graph weights, while the blue color indicates small weights.
### List of Tables

2.1 Average accuracies (with standard deviations) of different graphs for clustering on four databases. ................................................................. 25

2.2 Average normalized mutual information (NMI) of different graphs with standard deviations for clustering on four databases. ........................................ 27

2.3 Average accuracies of different graphs with standard deviations combined with the GHF label propagation method under different percentages of labeled samples (shown in the parenthesis) on YaleB and PIE databases. ......................... 28

2.4 Average accuracies of different graphs with standard deviations combined with the GHF label propagation method under different percentages of labeled samples (shown in the parenthesis) on ORL and USPS databases. ....................... 28

2.5 Average accuracies of different graphs with standard deviations combined with the semi-supervised discriminant analysis (SDA) method under different percentages of labeled samples (shown in the parenthesis) on YaleB and PIE databases. ................... 30

2.6 Average accuracies of different graphs with standard deviations combined with the semi-supervised discriminant analysis (SDA) method under different percentages of labeled samples (shown in the parenthesis) on ORL and USPS databases. ..................... 31

2.7 Average time cost (seconds) on PIE database. ............................................. 33

3.1 Average recognition rates (%) with standard deviations of all compared methods on COIL object database. ................................................................. 47

3.2 \(p\)-value between SRRS and other methods on the COIL object database. The asterisk * indicates that the difference between Method A and Method B is statistically significant when \(p = 0.05\). .............................................................................. 48

3.3 Average recognition rates (%) on ALOI object database. ............................... 52

3.4 Average recognition rates (%) of all compared methods on YaleB and FERET face databases. ................................................................. 53

3.5 Recognition rates (%) of all compared methods on AR face databases. .......... 55

3.6 Verification rates (%) on UB KinFace database (5-fold cross validation). “C vs. Y” and “C vs. O” denote child-young parent verification and child-old parent verification, respectively. ................................................................. 57

4.1 Notations .................................................................................................. 63
4.2 Classification Accuracy (%) on UCI Daily Activity Dataset. \( N_{tr} \) is the number of training samples randomly chosen from each activity. V1 and V2 denote the View-1 and View-2, respectively.

4.3 Classification accuracy (%) on Multimodal Spoken Word dataset. \( N_{tr} \) is the number of training samples randomly chosen from each word. V1, V2 and V3 denote the View-1, View-2 and View-3, respectively. SV and MV denote the single-view and multi-view methods, respectively.

4.4 Fusion results on UCI Daily and Sports Activity dataset when \( N_{tr} = 10 \).

5.1 Clustering accuracies with standard derivation and running time of all compared methods on Keck dataset.

5.2 Clustering accuracies (with standard derivation) and running time of all compared methods on Weizmann dataset.

5.3 Clustering accuracies (%) with standard derivation of all compared methods on MAD dataset.

5.4 Normalized mutual information (NMI) of all compared methods on MAD dataset.

5.5 Clustering accuracies of TSC and the state-of-the-art motion segmentation methods on three datasets.

6.1 Notations.

6.2 Target datasets for experiments.

6.3 Subspace clustering accuracies (%) of all compared methods. The version A of RDSTL and our method uses 1000 auxiliary images, and version B uses 3000 images.

6.4 NMI of all compared methods. The version A of RDSTL and our method uses 1000 auxiliary images, and version B uses 3000 images.

6.5 Average classification accuracies (%) of all compared methods on three datasets.
Acknowledgments

First, I would like to express my deepest and sincerest gratitude to my advisor, Professor Yun Fu, for his constant guidance, advice, effort, patience, and encouragement during the past five years. I am grateful to him for being always supportive when I was in need, and I am truly fortunate to have him as my advisor. This dissertation would not have been possible without his tremendous help.

I would also like to thank my committee members, Professor Jennifer G. Dy and Professor Lu Wang, for their valuable time, insightful comments and suggestions. I am also thankful to Professor Yizhou Sun for her guidance and help.

In addition, I would like to thank all the members of the SMILE Lab, especially my coauthors and collaborators: Dr. Kang Li, Dr. Ming Shao, Dr. Yu Kong, Dr. Chengcheng Jia, Zhengming Ding, Handong Zhao, Shuyang Wang, Hongfu Liu, Joesph Robinson, Zhiqiang Tao, and Kai Li.

This dissertation would not have been possible without my family’s love and support. I would like to express my gratitude to my parents, my parents-in-law, and my younger sister. Last but not least, I especially thank my wife, Shumin Jing, for her unconditional support and encouragement.
Abstract of the Dissertation

ROBUST DATA REPRESENTATIONS FOR VISUAL LEARNING

by

SHENG LI

Doctor of Philosophy in Computer Engineering
Northeastern University, April 2017
Dr. Yun Fu, Advisor

Extracting informative representations from data is a critical task in visual learning applications, which mitigates the gap between low-level observed data and high-level semantic knowledge. Many traditional visual learning algorithms pose strong assumptions on the underlying distribution of data. In practice, however, the data might be corrupted, contaminated with severe noise, or captured by different types of sensors, which violates these assumptions. As a result, it is of great importance to learn robust data representations that could effectively and efficiently handle the noisy visual data.

Recent advances on low-rank and sparse modeling have shown promising performance on recovering clean data from noisy observations, which motivate us to develop new models for robust visual learning. This dissertation focuses on extracting mid-level feature representations from visual data such as images and videos. The research goals of this dissertation are twofold: (1) learning robust data representations from visual data, by exploiting the low-dimensional subspace structures; (2) evaluating the performance of the learned data representations on various analytics tasks of images and videos.

Three types of data representations are studied in this dissertation, including graph, subspace, and dictionary. First, two novel graph construction schemes are proposed, by integrating the low-rank modeling with graph sparsification strategies. Each sample is represented in the low-rank coding space. And it is revealed that the similarity measurement in the low-rank coding space is more robust than that in the original sample space. The proposed graphs could greatly enhance the performance of graph based clustering and semi-supervised classification. Second, low-dimensional discriminative subspaces are learned in single-view and multi-view scenarios, respectively. The single-view robust subspace discovery model is motivated from low-rank modeling and Fisher criterion, and it is able to accurately classify the noisy images. The multi-view subspace learning model is designed for extracting compact features from multimodal time series data, which leverages
a shared latent space and fuses information from multiple data views. Third, dictionary serves as expressive bases for characterizing visual data. A non-negative dictionary with Laplacian regularization is learned to extract robust features from human motion videos, which leads to promising motion segmentation results. In addition, a robust dictionary learning method is designed to transfer knowledge from source domain to a target domain with limited training samples.

In summary, this dissertation aims to address the challenges in processing noisy visual data captured in real world. The proposed robust data representations have shown promising performance in a wide range of visual learning tasks, such as image clustering, face recognition, human motion segmentation, and multimodal classification.
Chapter 1

Introduction

1.1 Background

High-dimensional and large-scale visual data are everywhere. Nowadays, consumer devices with high-definition (HD) cameras enable people to capture images or videos in various scenarios, which leads to a huge amount of real-world visual data. According to some statistics on social networks, every minute, over 400 hours of videos are uploaded to YouTube, and more than 50,000 photos are presented to Instagram. Meanwhile, large-scale visual data could also be observed in many other domains, such as medical imaging, video surveillance, etc. To deal with the high-dimensional and large-scale visual data, intelligent and efficient visual learning algorithms are highly desired. In the past decades, computer vision and machine learning researchers have made significant progress toward modeling visual data for diverse purposes. Successful real-world applications of visual learning include face recognition, object detection, video segmentation, etc.

One critical task in visual learning is to extract informative representations from data, in order to mitigate the gap between low-level observed data and high-level semantic knowledge. Many traditional visual learning algorithms pose strong assumptions on the underlying distribution of data. However, the visual data captured in real-world might be corrupted, contaminated with severe noise, which violates the underlying assumptions. As a result, the system performance would be heavily degraded.

In this dissertation, we present a set of novel approaches, which could effectively and efficiently handle the noisy visual data by learning robust data representations. In particular, we consider the noisy visual data in a general sense. The noise might be: (1) Gaussian noise; (2) Random corruptions; (3) Missing values in data, due to data loss during transmission or other reasons;
CHAPTER 1. INTRODUCTION

(4) Outliers or anomalies; (5) Uncertainty within one modality; (6) Uncertainty across multiple modalities. The first four are well aligned with the traditional interpretations of noise, while the last two are considered as special cases of noise. In common settings of visual learning, one object is usually corresponding to multiple instances, such as multiple face images of the same person. If the multiple instances are from the same modality, variations on appearance may introduce uncertain information. For example, face images from the same person may have expression variations or lighting changes. In addition, if the object is captured as multiple modalities using different sensors, the variations across different modalities would introduce another level of uncertainty. We aim to design novel visual learning algorithms that could deal with multiple types of noise.

It has been extensively demonstrated that exploiting the low-dimensional structure from high-dimensional data will greatly benefit the visual learning tasks. Recent advances on low-rank and sparse modeling have shown promising performance on recovering clean data from noisy observations, by discovering the low-dimensional subspace structures. This observation motivate us to develop new models for robust visual learning. This dissertation focuses on extracting mid-level feature representations from visual data such as images and videos. The research goals of this dissertation are twofold: (1) learning robust data representations from visual data, by exploiting the low-dimensional subspace structures; (2) evaluating the performance of the learned data representations on various analytics tasks of images and videos.

Three types of data representations are studied in this dissertation, including graph, subspace, and dictionary. First, two novel graph construction schemes are proposed, by integrating the low-rank modeling with graph sparsification strategies. Each sample is represented in the low-rank coding space. And it is revealed that the similarity measurement in the low-rank coding space is more robust than that in the original sample space. The proposed graphs could greatly enhance the performance of graph based clustering and semi-supervised classification. Second, low-dimensional discriminative subspaces are learned in single-view and multi-view scenarios, respectively. The single-view robust subspace discovery model is motivated from low-rank modeling and Fisher criterion, and it is able to accurately classify the noisy images. The multi-view subspace learning model is designed for extracting compact features from multimodal time series data, which leverages a shared latent space and fuses information from multiple data views. Third, dictionary serves as expressive bases for characterizing visual data. A non-negative dictionary with Laplacian regularization is learned to extract robust features from human motion videos, which leads to promising motion segmentation results. In addition, a robust dictionary learning method is designed to transfer knowledge from source domain to a target domain with limited training samples.
CHAPTER 1. INTRODUCTION

From the perspective of machine learning task, this dissertation involves clustering, semi-supervised learning, classification, multi-view learning, time-series modeling, graph mining, subspace learning, dictionary learning, and transfer learning. The proposed approaches have obtained remarkable improvements on many real-world applications, including image clustering, image classification, face recognition, kinship verification, and human motion segmentation.

1.2 Related Work

1.2.1 Subspace Learning

Subspace learning is an effective technique in extracting informative features from data, which reduces the dimensionality of data through linear or nonlinear projections. It has been extensively studied and widely used in many real-world applications, such as face recognition, object recognition and visualization. The basic idea of subspace learning methods is to project high-dimensional samples into a low-dimensional subspace, in which some specific properties could be satisfied. According to the availability of class labels, subspace learning methods can be mainly divided into three groups: unsupervised methods, supervised methods, and semi-supervised methods.

The unsupervised methods only utilize unlabeled data [1], semi-supervised methods make use of the partial labeled data [2], and supervised methods learn subspaces using the fully labeled data [3, 4]. The representative unsupervised methods include principal component analysis (PCA) [1] and locality preserving projections (LPP) [5]. PCA projects data into a low-dimensional subspace by maximizing the variance of data. LPP is a non-parametric method, which preserves the neighborhood structure of samples on manifold.

Supervised subspace learning methods are very effective in extracting discriminative features, and usually achieve promising performance in classification tasks. Linear discriminant analysis (LDA) [6] is developed upon the Fisher criterion, which aims at finding a projection to maximize the inter-class scatter and minimize the intra-class scatter simultaneously. Many supervised subspace methods have been proposed to improve LDA. Local Fisher discriminant analysis (LFDA) [7] uses local neighborhood information to construct the weighted between-class and within-class scatter matrices, and then performs discriminant analysis. Subclass discriminant analysis [8] models the data using mixture of Gaussians, and redefines the scatter matrices used in LDA. LSDA [9] preserves both discriminant and local geometrical structure in data. Those methods usually obtain promising results on clean data, since they place specific assumptions on data.
distributions. However, when the data are corrupted by large amount of noise or large variations in real applications, these assumptions may be invalid, and the noise or variation can reduce the separability in a classification task. Therefore, the performance is heavily degraded. In addition, traditional methods require the vectorized data as input, while some advanced methods learn bilinear projections that directly process high-order data (e.g., images or EEG signals) without vectorization [10, 11].

Semi-supervised subspace learning methods make use of both labeled data and unlabeled data. The most representative method is semi-supervised discriminant analysis (SDA) [12], which employs a graph based smoothness regularity term to extend the objective function of linear discriminant analysis (LDA) [6].

1.2.2 Subspace Clustering

Subspace clustering is an effective technique which can automatically group the samples into low-dimensional subspace. It has achieved impressive performance in many real-world applications, such as motion segmentation [13], face clustering [14] and digit clustering [15]. Sparse subspace clustering (SSC) [14] enforces a sparse constraint on the coefficients. Low-rank representation (LRR) [13] considers the global structure of sample space, and usually achieves better performance than LRR. Least-square regression (LSR) [16] is very efficient by using Frobenius norm. Sparse additive subspace clustering (SASC) extends SSC to the additive nonparametric setting [17]. Discriminative subspace clustering (DSC) [18] incorporates discriminative information into the model. Smooth representation (SMR) makes use of the grouping effect to further enhance the subspace clustering performance [19]. In addition, many algorithms have been devised to reduce the computational cost of subspace clustering [20, 21, 15].

1.2.3 Low-Rank Modeling

Low-rank modeling is becoming popular and practical recently [22], due to its successful applications in many fields, such as data compression [23], subspace clustering [24, 25], image processing [26, 27] and multimedia analysis [28]. Robust PCA [23] is a representative low-rank modeling method. Given an observed and usually corrupted sample set \(X_O\), Robust PCA decomposes \(X_O\) into a low-rank, clean sample set \(X_L\) and a sparse, noisy sample set \(E\), i.e., \(X_O = X_L + E\). It shows impressive performance in background modeling and shadow removal. One major assumption in RPCA is that data are drawn from a single subspace. In practice, the underlying structure of
CHAPTER 1. INTRODUCTION

data could be multiple subspaces. Low-Rank Representation (LRR) is designed to find underlying
structures of noisy data [24].

Given a sample set \( X = [x_1, x_2, \ldots, x_n] \), The objective function of LRR is as follows

\[
\begin{align*}
\min_{Z,E} & \quad \text{rank}(Z) + \lambda_1 \|E\|_0 \\
\text{s.t.}, & \quad X = XZ + E,
\end{align*}
\]

where \( \text{rank}(\cdot) \) denotes the rank function, \( Z \in \mathbb{R}^{N \times N} \) is the low-rank coding matrix for \( X \), \( E \in \mathbb{R}^{d \times N} \) is the reconstruction error matrix, \( \|E\|_0 \) denotes the \( l_0 \) norm of matrix \( E \), and \( \lambda_1 \) is a trade-off parameter. The above problem is very difficult to solve due to the non-convexity of rank function and \( l_0 \) norm. Usually, they can be converted to trace norm (i.e., nuclear norm) and \( l_1 \) norm, respectively, and then numerous optimization algorithms can be applied to solve the problem.

LRR may suffer from two problems. The first one is insufficient data sampling since LRR simply uses the data matrix itself as the basis for representation. Second, the optimization of LRR requires multiple SVD calculations that are very time consuming. In [29], LatLRR is proposed to solve the insufficient sampling problem by considering the effects of hidden data for representation. In addition, active subspace [30] and Divide-Factor-Combine LRR (DFC-LRR) [31] employ various matrix factorization algorithms to tackle the above problems. Recently, a structured low-rank representation method [32] is proposed for image classification. A unified multi-scale low-rank representation approach is designed for image segmentation [33]. The low-rank constraint can also be employed to learn robust subspace [34], to construct reliable graphs [35], to learn effective online metrics [36], or to detect outliers in multi-view settings [37].

DLRD [38] is a low-rank dictionary learning method, which introduces low-rank constraints on the sub-dictionaries for each class, and performs sparse representation for face recognition. The learned dictionary in DLRD is low-rank and discriminative, which is beneficial for classification tasks. Nevertheless, the testing stage of DLRD is very time consuming, as it has to calculate sparse coefficients for every test sample. This is also a key difference between DLRD and our approach, since we perform classification on subspace that is very efficient. In [39], a low-rank method with structural incoherence is applied to face recognition. It first decomposes raw images into low-rank part and sparse part, and then applies PCA on the low-rank part to obtain a subspace. Finally, it employs sparse representation for classification. They did not, however, learn the low-rank representation and a discriminative subspace simultaneously. In this manner, the low-rank part is expected to be discriminative and benefit classification tasks. In [32], a structured low-rank representation method is presented for image classification. In [40], a LRR-based discriminative projection method
CHAPTER 1. INTRODUCTION

(LRR-DP) is proposed for feature extraction. It first applies LRR to recover the data matrix, and then finds a discriminative projection by designing a criterion that incorporates both clean data and noise. In this case, LRR is regarded as a data pre-processing method, and is performed only once to decompose sample set into two parts, the low-rank denoised samples and associated sparse noise. However, this decomposition is not guaranteed to be optimal for classification, as it doesn’t make use of any class prior information. In [41], a discriminant regularization term is incorporated into the formulation of Robust PCA. It separately learns low-rank data representation and subspace, which means the obtained subspace cannot be guaranteed to be optimal.

1.2.4 Multi-View Learning

Multi-view learning has been receiving increasing attention in recent years. One implicit assumption is that either view alone has sufficient information about the samples, but the learning complexity can be reduced by eliminating hypotheses from each view if different views contribute diverse information [42]. Multi-view learning has been widely applied to many problems, such as clustering [43, 44], classification [45, 46], semi-supervised learning [47, 48], person re-identification [49], and outlier detection [37].

Projecting data collected from multiple views onto a shared subspace is considered as an effective strategy in multi-view learning. The classical method, canonical correlation analysis (CCA) [50], projects two sets of observations onto a subspace by maximizing their correlations, which has been extended to multiple views [51, 52]. Most recently, Ding et al. incorporated low-rank constraints in learning common subspace for multi-view data. Kan et al. extended the linear discriminant analysis method to multi-view setting [46] and obtained impressive performance on image classification.

1.2.5 Self-Taught Learning

In many real-world visual learning tasks, the assumption of sufficient training data may not always hold. Thus, involving additional data resources effectively to overcome the shortage of training data becomes an important problem. Most representative solutions include semi-supervised learning [53] and transfer learning [54]. The former solution addresses this problem by using a large amount of unlabeled data from the same domain with same distribution to build better classifiers, while the latter one tries to leverage labeled data from related homogenous tasks. However, neither unlabeled data with same distribution nor labeled data from homogenous tasks are easy to get.
CHAPTER 1. INTRODUCTION

Recently, there has been a surge of interest in the topic of self-taught learning by involving unlabeled data without the above restrictions [55, 56, 57, 58, 59]. Raina et al. first proposed the concept of self-taught learning by applying sparse coding mechanism to construct a higher-level representation from the unlabeled data [55, 58]. Lee et al. extended Raina’s work by presenting a generalization of sparse coding module which could be suited to model other data types drawn from any exponential family distribution [56]. From the application point of view, Dai et al. proposed a clustering algorithm in the spirit of self-taught learning by allowing the feature representation from the auxiliary data to influence the target data through a common set of features [57]. Kuen et al. employed the core idea of self-taught learning, and transferred stacked auto encoders for visual tracking [60]. However, existing self-taught learning methods do not take advantage of any global structure information in the target set, as they encode each input signal independently. Besides, a generalizable schema of self-taught learning for both supervised and unsupervised learning tasks has not been well studied yet.

1.3 Dissertation Organization

The rest of this dissertation is organized as follows.

Chapter 2 presents a robust graph construction approach. Graphs have been widely applied in modeling the relationships and structures in real-world applications. By virtue of recent advances in low-rank subspace recovery, we notice that the similarity between every two samples evaluated in the low-rank coding space is more robust than that in the sample space. Based on the low-rank codings, we propose two graph construction methods that incorporate the $k$-NN constraint and $b$-matching constraint, respectively. Extensive evaluations on several benchmark databases demonstrate the superiority of the proposed graphs over several state-of-the-art graphs in data clustering, transductive and inductive semi-supervised learning.

Chapter 3 presents a robust subspace discovery approach. Subspace learning is widely used in extracting discriminative features for classification. However, conventional subspace learning methods usually have strong assumptions on the data distribution, and therefore they are sensitive to the noisy data. The learned subspace has limited discriminability. To address this problem, we propose to exploit a discriminative and robust subspace, which is insensitive to noise or pose/illumination variations, for dimensionality reduction and classification [35]. Our approach achieves promising performance on noisy image classification and noisy face recognition.
CHAPTER 1. INTRODUCTION

Chapter 4 presents a robust multi-view subspace learning approach. In particular, we focus on learning discriminative features for classifying multi-view multivariate time series (m.t.s.) data. Our approach keeps the original temporal structure of m.t.s. data, and projects m.t.s. from different views onto a shared latent subspace. It also incorporates discriminative information by minimizing the within-class separability and maximizing the between-class separability of m.t.s. in the shared latent subspace. Moreover, a Laplacian regularization term is designed to preserve the temporal smoothness within m.t.s.. Remarkable performance are observed on two real-world datasets.

Chapter 5 presents a robust dictionary learning approach for human motion segmentation. By taking advantages of the robust dictionary learning, we propose a temporal subspace clustering (TSC) method for human motion segmentation [61]. A least-square regression based formulation is adopted to learn compact codings for each data point. To obtain more expressive codings, we learn a non-negative dictionary from data, instead of using the data self-expressive models. In addition, a temporal Laplacian regularization function to encode the sequential relationships in time series data. Experimental results on several human action and gesture datasets are reported and discussed.

Chapter 6 presents a robust dictionary learning approach for knowledge transfer. Self-taught learning is a special case of transfer learning, which transfers the useful knowledge learned from an auxiliary domain to help the learning tasks in target domain. We focus on building a self-taught coding framework, which can effectively utilize the rich low-level pattern information abstracted from the auxiliary domain, in order to characterize the high-level structural information in the target domain. By leveraging a high quality dictionary learned across auxiliary and target domains, the proposed approach learns expressive codings for the samples in the target domain. Extensive experiments on five benchmark datasets demonstrate the effectiveness of our approach.

Finally, Chapter 7 concludes this dissertation.
Chapter 2

Robust Graph Construction

2.1 Background and Motivation

Graph based data mining and machine learning has attracted increasing attention over the last decade, and many graph based learning algorithms have shown great success in various scenarios, such as classification, clustering, semi-supervised learning, and social network analytics. Graph provides a very effective way of representing underlying relationships in data. However, how to accurately measure these relationships during graph construction is always a challenging problem. On the other hand, sparsity in graphs is also preferred since sparse graphs have much less misconnections among dissimilar data points. In this chapter, we focus on addressing these two fundamental problems in graph construction, which are similarity metric and graph sparsification.

Based on spectral graph theory, graphs have been widely used in data clustering [62, 63, 64, 65]. Normalized cut is a representative graph based clustering method [66], and many other clustering algorithms have been presented, such as the constrained graph-based clustering [67], the bi-stochastic data similarity based clustering [68], and the efficient spectral clustering on graphs (ESCG) [69]. Another interesting and successful application of graph is semi-supervised learning (SSL). Due to the fact that unlabeled data are much easier to obtain than labeled ones, SSL has been extensively studied in recent years, as it learns from both labeled and unlabeled samples [70, 71]. Among various SSL techniques, graph based SSL (GSSL) always achieves inspiring performance on accuracy and speed [72, 73, 74, 75, 76, 77]. Zhu et al. [78] proposed to use the harmonic property of Gaussian random field over the graph for SSL. Zhou et al. [79] performed SSL with the local and global consistency (LGC). Wang et al. [80] proposed a linear neighborhood propagation (LNP) method that considers the linear neighborhood around each data point when constructing the graph. He et
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Similarity Metric
Complete Graph Construction

Low-Rank Representations

Input Images

Sparsification
Model-I: k-NN
Unbalanced Sparse Graph: Each node selects 2 nearest neighbors.

Model-II: b-Matching
Balanced Sparse Graph: The degree of each node $b = 2$.

Applications
Unsupervised Clustering
Spectral clustering

Transductive Semi-supervised Classification
Gaussian Harmonic Function (GHF)

Inductive Semi-supervised Learning
Semi-supervised Discriminant Analysis

Figure 2.1: Flowchart of our framework. It contains three parts: similarity metric, graph sparsification and applications. (1) Low-rank representation coefficients are utilized to measure the similarities between all pairs of samples, and a complete graph can then be built, where the bold edges imply larger graph weights. (2) $k$-NN and $b$-matching constraints are employed for graph sparsification. The first two parts are learnt iteratively. (3) The applications based on our graph include clustering, transductive and inductive semi-supervised classification.

al. [81] developed a generative model for GSSL by estimating priors and conditional probabilities. The similarity and dissimilarity are incorporated for graph based semi-supervised classification [82]. Cai et al. [12] proposed a semi-supervised discriminant analysis (SDA) method, which employs a graph based smoothness regularization term to extend the objective function of linear discriminant analysis (LDA) [6]. Ni et al. [83] designed a propagable graph for semi-supervised classification and regression. In addition, a generalized optimization framework for graph-based semi-supervised learning was developed in [84].

Although many graph based machine learning algorithms have been proposed, limited research has focused on how to construct effective graphs [85, 86, 80]. $k$-nearest neighbor ($k$-NN) and $\epsilon$-neighborhood are two popular and efficient graph construction schemes. However, neither methods generate graphs that are balanced or regular. Unbalanced (or irregular) graphs usually hinder learning performance, because the high degree nodes may be dominant in the learning process. $b$-matching method can solve this problem by learning a symmetric and balanced graph [87]. It enforces that the degree of every node in the graph is exactly equal to a constant $b$. Unfortunately, these resulting graphs are highly dependent on the similarity function, as the similarity values (e.g., graph weights) are very sensitive to sample variation or noise. In unsupervised and semi-supervised
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

learning, the algorithms usually show effective performance on data that obey the smoothness, cluster or manifold assumptions [88, 89, 90]. However, when the samples contain noise or large variations, these assumptions are often violated, and therefore the traditional similarity metrics (e.g., Gaussian function) often fail.

To address those limitations in existing methods, in this chapter, we design a novel graph construction approach to learn unbalanced or balanced graphs via a robust similarity metric. Recent advances on low-rank matrix recovery suggest that noisy data drawn from different subspaces can be correctly recovered by seeking the lowest-rank encoding matrix for all samples [91, 92]. Inspired by this observation, we propose to measure the similarities between different samples in the low-rank encoding space, instead of original sample space. Another observation is that a sparse graph can greatly improve the learning performance, such as the label propagation procedure in GSSL. The $k$-NN sparsification can provide fast solutions to constructing a sparse graph, and $b$-matching constraint is a necessary requirement for generating a balanced sparse graph that is more desired for learning tasks. These observations motivate us to jointly learn the low-rank codes and a balanced (or unbalanced) graph simultaneously. Fig. 2.1 illustrates our framework. In particular, two models are built by incorporating $k$-NN constraint and $b$-matching constraint, respectively.

Instead of using the biased estimators adopted in existing low-rank matrix recovery methods, we first relax the matrix rank and $l_0$ norm in our models by two unbiased estimators, matrix $\gamma$-norm and minimax concave penalty (MCP) norm, respectively, and then design a majorization-minimization augmented Lagrange multiplier (MM-ALM) algorithm to solve the proposed non-convex models. We show that this novel optimization algorithm can better recover the low-rank subspace structures that results in robust similarities during graph construction. After applying a graph re-weighting strategy, our graph can be used in many unsupervised and semi-supervised learning scenarios including data clustering, transductive and inductive semi-supervised classification. Experimental results on the Extended YaleB, PIE, ORL and USPS image databases demonstrate the effectiveness of our graph, compared with several state-of-the-art graphs.

In summary, our contributions include:

- We have proposed a new similarity metric based on low-rank subspace recovery. The $k$-NN and $b$-matching constraints are incorporated into the low-rank learning models to learn unbalanced or balanced graphs, respectively.

- We have relaxed the matrix rank and $l_0$ norm in low-rank representation model by using the matrix $\gamma$-norm and matrix MCP norm, respectively.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

- We have designed a non-convex optimization algorithm to solve the proposed models, and have shown its convergence properties.

- Besides transductive semi-supervised classification evaluated in [93], we have also extended the applications of our model to data clustering and inductive semi-supervised learning \(^1\). Note that we also compare the proposed models with [93] in this chapter.

2.2 Existing Graph Construction Methods

Constructing an effective graph is the most important component in graph based learning algorithms. As we discussed above, there has been some research specifically that targeted graph construction. In this section, we give a brief review of these related works sorted by unbalanced graphs and balanced graph, sparse representation based graphs, and low-rank learning based graphs.

2.2.1 Unbalanced Graphs and Balanced Graph

The \(k\)-NN graph and \(\epsilon\)-neighborhood graph are two representative unbalanced graphs. In the construction procedures, kernel function \(k(\cdot)\) (e.g., Gaussian kernel) is usually used to estimate the similarity between samples. Based on the similarity matrix, \(k\) significant neighbors for each node are greedily selected to construct a \(k\)-NN graph [66]. However, \(k\)-NN graph is unbalanced, as it always requires a symmetrization process after selecting \(k\) neighbors. An unbalanced graph may lead to a performance degradation in learning algorithms. In the \(\epsilon\)-neighborhood graph, each node is connected to those nodes within the distance \(\epsilon\). The linear neighborhood propagation (LNP) method improves the similarity measurement in \(k\)-NN graph and \(\epsilon\)-neighborhood graph, by utilizing the manifold information [80]. But, both \(\epsilon\)-neighborhood graph and LNP based graph are still unbalanced.

To address this unbalanced problem, Jebara et al. proposed a \(b\)-matching method to construct a balanced graph [87]. They employed a \(b\)-matching constraint during graph construction, and therefore the degree of every node is exactly equal to the constant \(b\). This method simultaneously ensures both symmetry and balance of the graph in learning procedures. However, the performance of a \(k\)-NN graph and a \(b\)-matching graph is highly dependent on the similarity metric. If the metric is sensitive to sample variation or noise in the data, the performance greatly reduced.

\(^1\)Semi-supervised learning can be either transductive or inductive. Transduction model only works on the labeled and unlabeled training samples, and it cannot deal with unseen data. Inductive model can naturally handle unseen data [70, 94].
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

2.2.2 Sparse Representation based Graphs

Another class of graph construction methods focuses on weighting the graphs. Sparsity could successfully recover signals in noisy scenarios [95], and several graph construction methods in which the sparse representation coefficients are employed as graph weights have been proposed [96, 97, 98, 99]. Yan et al. proposed an $l_1$ graph, which adopts sparse representation coefficients as the weights on a graph [96]. $l_1$ graph is constructed in a parameter-free manner, since the adjacency structure and graph weights are determined by the $l_1$ optimization algorithm automatically. Cheng et al. proposed a sparsity induced similarity (SIS) measure for label propagation in GSSL [100], which utilizes the coefficients of sparse decompositions. Furthermore, He et al. presented a non-negative sparse probability graph (SPG) to further improve the classification performance of SSL [97]. Nie et al. designed an iterative algorithm to solve the $l_1$ norm of spectral embedding minimization problem for semi-supervised classification [98]. Recently, the neighborhood graph construction problem was also modeled as a sparse coding problem with the locality constraint [99].

However, the reason for using sparse representation coefficients as graph weights is not quite clear. Especially when the labeled samples are very limited, sparse coefficients are very sensitive to the variation in labeled samples, and then the label propagation results would be affected negatively. In addition, all these sparse graphs are not balanced.

2.2.3 Low-Rank Learning based Graphs

Low-rank matrix recovery has attracted increasing attention in recent years [22], and it has been successfully applied to many areas [23, 91, 101, 34, 102]. Low-Rank Representation (LRR) [91, 92] was proposed to recover multiple subspaces in the presence of noise, and it has shown considerable effectiveness in subspace segmentation. To capture the global structure of sample set $X$, LRR determines a representation $Z$ by minimizing the rank of $Z$ \( \min_Z \text{rank}(Z) \) with a constraint $X = XZ$. This problem is NP-hard, however, it can be transformed into an equivalent convex optimization problem using nuclear norm $\| \cdot \|_*$ instead of $\text{rank}(\cdot)$ function. One drawback of LRR is that the nuclear norm and $l_{2,1}$ norm are biased estimators since they over-penalizes large singular values and large entries, respectively [103]. In this chapter, we present an effective solution to this problem by introducing the matrix $\gamma$-norm, which is a non-convex relaxation of matrix rank.

Recently, some graph construction methods that use low-rank representation coefficients as graph weights have been developed [104, 105, 106, 107, 108]. Among them, non-negative low-rank and sparse (NNLRS) graph [104] is the most relevant work to ours. NNLRS enforces low-rankness
and sparsity simultaneously, and employs the low-rank representation coefficients as graph weights. There are several key differences between NNLRS and our approach. First, NNLRS directly utilizes low-rank representation coefficients as graph weights, our approach, on the other hand, employs low-rank coefficients of each pair of samples to calculate their similarity. Second, the optimization techniques are different since we build non-convex optimization models. Third, our approach jointly learns a sparse and balanced graph with \( b \)-matching constraint, while NNLRS graph is unbalanced, which is undesired for semi-supervised classification.

### 2.3 Low-Rank Coding based Unbalanced Graph Construction

In this section, we first describe the motivation of designing robust similarity metric using low-rank codings. We then build an optimization model to learn the similarity matrix, and develop a non-convex optimization algorithm to solve this model.

#### 2.3.1 Motivation

Given a sample set \( X = [x_1, x_2, \ldots, x_n] \) (each column is a sample, \( x_i \in \mathbb{R}^d \)), graph construction models aim at building a weighted graph \( G \). Typically, there are three steps: similarity measurement between each pair of samples, sparsification of the graph, and graph re-weighting. The output graph can be expressed as \( G = S \odot W \), where \( S \) is a binary matrix whose elements indicate the edges in graph, \( W \) is a similarity matrix or other user-defined weighting matrices, and \( \odot \) is the Hadamard product, i.e., \( G_{ij} = (S \odot W)_{ij} = S_{ij}W_{ij} \).

We focus on the first major step of graph construction: similarity measurement. Many current similarity metrics, such as Gaussian function, are sensitive to noise or large intra-class variations. Inspired by the low-rank subspace recovery theory, we propose to estimate similarity in the low-rank code space to address this problem. Let \( Z \) denote the unknown coefficient matrix for sample set \( X \), low-rank method learns a matrix \( Z \) which has the lowest \( \text{rank}(Z) \) and satisfies the constraint \( X = AZ + E \), where \( A \) is an over-complete dictionary and \( E \) is the sparse noise component [92]. The \( i \)-th column vector in \( Z \) is the low-rank code vector for \( x_i \). Due to the fact that \( Z \) can correctly recover multiple subspaces and its low-rankness, the low-rank codes in \( Z \) belonging to one subspace (i.e., samples in the same class or same cluster) should be highly correlated. Meanwhile, low-rank coefficients are very robust to different kinds of noises [92]. Thus, it is reasonable to employ low-rank codes to estimate the similarity between all pairs of samples.
To clearly illustrate our new similarity metric, Fig. 2.2 compares two similarity metrics of digits images in the original space and low-rank code space. It shows that, in case of large intra-class variation or large inter-class correlation, similarity values calculated in original space may be unreliable. However, the noise-insensitive low-rank codes can correctly recover the subspace structures of multiple classes, and low-rankness means that codes belonging to the same class should have high correlations. Thus, the similarity metric in low-rank space obtains better results as shown in Fig. 2.2.

Calculating the inner product of every pair of low-rank representation coefficient vectors is a straightforward way to obtain similarity value. After obtaining a fully connected similarity matrix $\hat{W}$, where $\hat{W}_{i,j} = |(Z^T Z)_{i,j}|$, we should perform the second step of graph construction: sparsification. $k$-NN is a simple yet effective strategy for sparsifying a graph. Thus, we propose to build an unbalanced sparse graph construction model (Model-I) by integrating the ideas of low-rank coding and $k$-NN sparsification.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

2.3.2 Problem Formulation

Based on the above observations, we propose an optimization model to learn low-rank codes and sparsify the graph using $k$-NN simultaneously. The objective function of our Model-I is:

$$
\begin{align*}
\min_{Z,E,S} & \quad \text{rank}(Z) + \lambda_1 \|E\|_0 - \lambda_2 \sum_{i,j=1}^{n} S_{ij}(Z^\top Z)_{ij} \\
\text{s.t.} & \quad X = AZ + E, \sum_{j=1}^{n} S_{ij} = k, S_{ii} = 0,
\end{align*}
$$

(2.1)

where $A$ is the dictionary with the size of $n \times a$, $Z$ is the low-rank coefficient matrix, $E$ is a sparse noise matrix, $\lambda_1$ and $\lambda_2$ are trade-off parameters to balance the effects of other terms, $S$ is a binary $k$-NN graph and $k$ is the number of nearest neighbors.

In Eq. (2.1), the first two terms $\text{rank}(Z) + \lambda_1 \|E\|_0$ denote the low-rank representation of noisy data matrix $X$. The last term indicates the $k$-NN sparsification, which means $k$ nearest neighbors are selected for each node.

Eq. (2.1) is a variant of rank minimization problem. Generally, like LRR, this kind of problem can be relaxed by using trace norm (nuclear norm) and $l_1$ norm (or $l_{2,1}$ norm), and then solved by some convex optimization tools, such as inexact augment Lagrange multiplier (ALM) algorithms [109] and linearized alternating direction method with adaptive penalty (LADMAP) [110]. However, Wang et al. noted that the nuclear norm and $l_1$ norm are actually biased estimators since they over-penalize large singular values and large entries [103]. They devised a matrix minimax concave penalty (MCP) norm and a matrix $\gamma$-norm to remodel a non-convex version of the matrix recovery problem that is originally defined in RPCA. Inspired by the effectiveness of this non-convex relaxation, we propose to reformulate the low-rank representation problem by using matrix MCP norm and matrix $\gamma$-norm.

The matrix MCP norm is defined as:

$$
M_{\lambda, \gamma}(A) = \sum_{i,j} \phi_{\lambda, \gamma}(A_{i,j}),
$$

(2.2)

where

$$
\phi_{\lambda, \gamma}(t) = \lambda \int_{0}^{t} [1 - \frac{x}{\gamma \lambda}]_+ dx = \begin{cases} 
\frac{\gamma \lambda^2}{2}, & \text{if } |t| \geq \gamma \lambda \\
\gamma |t| - \frac{t^2}{2\gamma}, & \text{otherwise.}
\end{cases}
$$

$[z]_+ = \max(z, 0)$. Here, we choose $\lambda = 1$, and denote $M_{\gamma}(A) = M_{1, \gamma}(A)$ for simplicity.

The matrix $\gamma$-norm is defined as [103]:

$$
\|A\|_{\gamma} = \sum_{i=1}^{r} \phi_{1, \gamma}(\sigma_i(A)) = M_{\gamma}(\sigma(A)), \quad \gamma > 1,
$$

(2.3)
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

where \( \sigma(A) = (\sigma_1(A), \cdots, \sigma_r(A))^T \) denotes a function from \( \mathbb{R}^{m \times n} \) to \( \mathbb{R}^r_+ \), \( r = \min(m, n) \). The matrix \( \gamma \)-norm is non-convex w.r.t \( A \).

Then, problem (2.1) can be reformulated as:

\[
\begin{align*}
\min_{Z,E,S} & \quad \|Z\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E) - \lambda_2 1_n^\top (S \circ (Z^\top Z)) 1_n \\
\text{s.t.} & \quad X = AZ + E, \sum_{j=1}^n S_{ij} = k, S_{ii} = 0,
\end{align*}
\]

(2.4)

where \( \circ \) is the Hadamard product.

2.3.2.1 Optimization

In this subsection, we relax the objective function in Eq. (2.4) and design a majorization-minimization (MM)-ALM algorithm to solve this problem.

The last term in Eq. (2.4), \( 1_n^\top (S \circ (Z^\top J)) 1_n \), makes it difficult to solve the objective function. Fortunately, we can relax it to \( 1_n^\top (S \circ (Z^\top J)) 1_n \) with a new constraint \( Z = J \). Then (2.4) can be rewritten as:

\[
\begin{align*}
\min_{Z,E,S,J} & \quad \|J\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E) - \lambda_2 1_n^\top (S \circ (Z^\top J)) 1_n \\
\text{s.t.} & \quad X = AZ + E, \sum_{j=1}^n S_{ij} = k, S_{ii} = 0, Z = J.
\end{align*}
\]

(2.5)

We first introduce the generalized singular value shrinkage operator \( S_{\tau,\Lambda} \) and generalized shrinkage operator \( D_{\gamma,W} \) [103]:

\[
S_{\tau,\Lambda} = U_X D_{\tau,\Lambda}(\Sigma_X)(V_X)^\top,
\]

(2.6)

\[
[D_{\gamma,W}(A)]_{ij} = \text{sgn}(A_{ij})(|A_{ij} - \tau W_{ij}|)_+,
\]

(2.7)

where \( \Sigma \) and \( \Lambda \) are non-negative matrices.

The MM-ALM algorithm consists of an outer loop and an inner loop. In each iteration, the outer loop replaces the non-convex problem by its locally linear approximation (LLA) to form a weighted convex problem, while an inner loop is an inexact ALM algorithm.

In the outer loop, we reformulate the objective function as follows. Since the objective function in Eq. (2.5) is concave w.r.t. \( (\sigma(J), |E|, |Z|, |S|) \), we can approximate it by the LLA, and obtain the following objective function:

\[
\begin{align*}
\min_{Z,E,S,J} & \quad f(J, E) = Q_{\gamma_1}(\sigma(J)|\sigma(J)|^{\text{old}} + \lambda_1 Q_{\gamma_2}(E|E|^{\text{old}}) \\
& - \lambda_2 1_n^\top (S \circ (Z^\top J)) 1_n \\
\text{s.t.} & \quad X = AZ + E, \sum_{j=1}^n S_{ij} = k, S_{ii} = 0, Z = J,
\end{align*}
\]

(2.8)
where
\[ Q_\gamma(A|A^{old}) = M_\gamma(A^{old}) + \sum_{i,j} (1 - |A_{ij}^{old}|/\gamma)_+ (|A_{ij}| + |A_{ij}^{old}|). \]
is the LLA of \( M_\gamma(A) \) given \( A^{old} \).

In the inner loop, we utilize the inexact ALM algorithm to solve Eq. (2.8) by alternately updating different sets of variables. First we update the variables \( J, Z \) and \( E \) when \( S \) is fixed, and then update \( S \) when others are fixed. The augmented Lagrange function is:
\[
L = Q_\gamma_1(\sigma(J)|\sigma(J)^{old}) + \lambda_1 Q_\gamma_2(E|E^{old}) - \lambda_2 1_n^T (S \circ (Z^T J)) 1_n + <X - AZ - E, Y_1 > \\
+ < Z - J, Y_2 > + < \sum_j S_{ij} - k, Y_3 > \\
+ \frac{\mu}{2} (||X - AZ - E||_F^2 + ||Z - J||_F^2),
\]
where \( Y_1, Y_2 \) and \( Y_3 \) are Lagrange multipliers and \( \mu > 0 \) is a penalty parameter.

In particular, we alternately update the variables \( J, Z, E \) and \( S \) in the \( k + 1 \) iteration as follows:
\[
J_{k+1} = S_{1/\mu, A}(Z_k + \frac{\lambda_2 S_k \circ Z + Y_2}{\mu}), 
\]
\[
Z_{k+1} = (I_n + A^T A)^{-1}(A^T X - A^T E_k + J_{k+1} + (A^T Y_1 - Y_2 + \lambda_2 S_k \circ J_{k+1})/\mu), 
\]
\[
E_{k+1} = D_{\lambda/\mu, W}(A - AZ_{k+1} + Y_1/\mu), 
\]
\[
S_{k+1} = \arg \min_{S_k} -\frac{\lambda_2}{\mu k} 1_n^T (S_k \circ (Z_{k+1}^T J_{k+1})) 1_n, 
\]
s.t. \( \sum_j (S_k)_{ij} = k, (S_k)_{ii} = 0. \)

Eq. (2.13) can be solved by a standard \( k \)-NN solver. The details of the algorithm is outlined in Algorithm 1. In this chapter, we use sample set \( X \) itself as dictionary, that is, \( A = X \).

**Lemma 1** When \( S \) is fixed, the objective function values of (2.10) obey
\[
f(J, E, Z) \leq Q_\gamma_1(\sigma(J)|\sigma(J)^{old}) + \lambda_1 Q_\gamma_2(E|E^{old}) - \lambda_2 1_n^T (S \circ (Z^T J)) 1_n \\
\leq Q_\gamma_1(\sigma(J)^{old}|\sigma(J)^{old}) + \lambda_1 Q_\gamma_2(E^{old}|E^{old}) - \lambda_2 1_n^T (S \circ (Z^{old} J^{old})) 1_n \\
= f(J^{old}, E^{old}, Z^{old}).
\]
This lemma can be proved using the Proposition 4 in [103]. It demonstrates the local convergence property of our algorithm.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

2.3.3 Complexity Analysis

Our optimization algorithm contains two parts, LLA and inexact ALM. As suggested in [103], we adopt the one-step LLA strategy, which runs the outer loop in Algorithm 1 only once, to alleviate the computational cost. In particular, the solutions of LRR are used for initializing the one-step LLA.

The computational complexity of Algorithm 1 is mainly dependent on the inner loop, where the most time-consuming parts are Steps 6-8. In Step 6, the SVT operator requires singular value decomposition of matrices of size $n \times n$, which costs $O(n^3)$, where $n$ is the total number of samples. In Step 7 and Step 8, the matrix inversion and matrix multiplication also cost $O(n^3)$. Assume that the number of iterations in Algorithm 1 is $l$, the overall computational complexity of this algorithm would be $O(ln^3)$. Thus, the scalability of Algorithm 1 is determined by the sample size $n$, like many other low-rank learning methods. [31] presents a distributed solution to improve the scalability, which would be adopted to accelerate our algorithm in the future work. We will show the running time of our algorithm and its competitors in the experiments.

2.3.4 Discussions

Traditionally, graph construction methods either utilize various similarity functions to estimate the weights [66], or leverage on the manifold information [80]. In our approach, we propose a new similarity metric by taking advantage of the subspace structure of sample set. The experimental results will demonstrate that exploiting such subspace structures in graph construction would greatly benefit the unsupervised and semi-supervised learning tasks.

Moreover, we unify two steps of graph construction, similarity measurement and sparsification, into a single framework. These two steps can guide each other in an iterative learning manner. Existing graph construction methods either follow the two-step strategy (e.g., $k$-NN graph, $\epsilon$-neighborhood graph), or estimate the weights and sparsify the graph at once (e.g., $l_1$ graph [96], LRR graph [92]). In our approach, we jointly learn the graph weights and sparsify the graph. Experiments will show the benefits of our pipeline.

2.4 Low-Rank Coding based Balanced Graph Construction

In this section, we explain why the $b$-matching constraint is necessary, and then use it to replace the $k$-NN sparsification in our Model-I. We build a joint optimization model Model-II to
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Algorithm 1. Solving Problem (2.5) via MM-ALM

Input: data matrix $X$, dictionary $A$, parameter $\lambda_1, \lambda_2$,  
$Z_0 \in \mathbb{R}^{a \times n}$, $J_0 \in \mathbb{R}^{a \times n}$, $E_0 \in \mathbb{R}^{d \times n}$, $S_0 \in \mathbb{R}^{n \times n}$,  
$Y_1 \in \mathbb{R}^{d \times n}$, $Y_2 \in \mathbb{R}^{a \times n}$, $\mu_0 = 10^{-3}$, $\mu_{\text{max}} = 10^5$,  
$\rho = 1.3$, $k = 0$, $j = 0$, $\epsilon = 10^{-5}$

Output: $S, Z, E$

1: while not converged do
2: Initialize $\mu_0 = 10^{-3}$, $Y_1$, $Y_2$, $k = 0$;
3: $\Lambda = \text{Diag}(1_n - \sigma(J^j)/\gamma_1)^+$; 
4: $W = (1_m 1_n^T - |S^j|/\gamma_2)^+$; 
5: while not converged do
6: Update $J_{k+1}^{j+1}$ using (2.10), given others fixed; 
7: Update $Z_{k+1}^{j+1}$ using (2.11), given others fixed; 
8: Update $E_{k+1}^{j+1}$ using (2.12), given others fixed; 
9: Update $S_{k+1}^{j+1}$ using (2.13), given others fixed; 
10: Symmetrize $S_{k+1}^{j+1} = \max(S_{k+1}^{j+1}, S_{k+1}^{j+1T})$; 
11: Update the multipliers $Y_1$ and $Y_2$ 
   $Y_1 = Y_1 + \mu_k (X - AZ_{k+1}^{j+1} - E_{k+1}^{j+1})$, 
   $Y_2 = Y_2 + \mu_k (J_{k+1}^{j+1} - Z_{k+1}^{j+1})$; 
12: Update the parameter $\mu_{k+1}$ by 
   $\mu_{k+1} = \min(\rho \mu_k, \mu_{\text{max}})$ 
13: Check the convergence conditions 
   $\|X - AZ - E\|_\infty < \epsilon$ and $\|J - Z\|_\infty < \epsilon$. 
14: $k = k + 1$; 
15: $j = j + 1$; 
16: end while 
17: end while

learn the similarity matrix and the balanced graph simultaneously.

2.4.1 Motivation and Formulation

Sparsity of graph is an important requirement for ensuring the effectiveness of learning algorithms, since sparse graphs have much less misconnections among dissimilar samples. For example, in graph based semi-supervised classification, sparsity helps improve the classification accuracy and computational performance [85]. As we discussed above, $k$-NN and $\epsilon$-neighborhood are two commonly used strategies to generate a sparse graph, and $l_1$ graphs also meet the sparsity requirement in nature. However, all these graphs are unbalanced, which hinders learning performance, since some high degree nodes may dominate the learning results. To address this problem, we incorporate the $b$-matching constraint that is designed for learning a balanced graph.
In particular, we replace the $k$-NN sparsification with $b$-matching constraint in Model-I, and then build Model-II as:

$$
\min_{Z,E,S,J} \|J\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E) - \lambda_2 1_n^T (S \circ (Z^T J)) 1_n \\
\text{s.t. } X = AZ + E, \sum_{j=1}^n S_{ij} = b, S_{ij} = S_{ji}, Z = J,
$$

(2.14)

where $A$ is the dictionary, $Z$ is the low-rank coefficient matrix, $E$ is a sparse noise matrix, $\lambda_1$ and $\lambda_2$ are trade-off parameters to balance the effects of other terms, $S$ is a binary balanced graph and $b$ is the constraint on the degree of each node.

Eq. (2.14) differs from Eq. (2.5) in the constraints. The last term in Eq. (2.14) $(-\lambda_2 1_n^T (S \circ (Z^T J)) 1_n)$ and the $b$-matching constraint $\sum_{j=1}^n S_{ij} = b$ indicate that each node should have a degree of $b$, while the total similarity value should be maximized.

### 2.4.2 Optimization

In this subsection, we extend the optimization algorithm developed in Section 2.3 to solve Eq. (2.14).

The outer loop for Model-II is the same as that for Model-I, and the inner loop can also be solved by inexact ALM algorithm. The augmented Lagrange function is:

$$
L = Q_{\gamma_1} (\sigma(J) | \sigma(J)^{old}) + \lambda_1 Q_{\gamma_2}(E|E^{old}) - \lambda_2 1_n^T (S \circ (Z^T J)) 1_n + <X - AZ - E, Y_1 > \\
+ <J - Z, Y_2 > + < \sum_{j} S_{ij} - b, Y_3 > \\
+ \frac{\mu}{2} (\|X - AZ - E\|_F^2 + \|J - Z\|_F^2),
$$

(2.15)

where $Y_1, Y_2$ and $Y_3$ are Lagrange multipliers and $\mu > 0$ is a penalty parameter.

In particular, we alternately update the variables $J$, $Z$, $E$ and $S$ in each iteration. The solvers for $J$, $Z$ and $E$ have been shown in Eqs. (2.10), (2.11) and (2.12). The subproblem for solving $S$ is:

$$
S_{k+1} = \arg \min_{S_k} -\frac{\lambda_2}{\mu_k} 1_n^T (S_k \circ (Z_{k+1}^T J_{k+1})) 1_n. \\
\text{s.t. } \sum_{j} (S_k)_{ij} = b, (S_k)_{ij} = (S_k)_{ji}.
$$

(2.16)

Eq. (2.16) can be solved by a fast $b$-matching algorithm that has been recently proposed in [111]. The time complexity of $b$-matching solver is $O(n^{2.5})$ [111]. One can modify Algorithm 1 to solve (2.14), by replacing (2.13) with (2.16) in the $9^{th}$ step and removing the $10^{th}$ step.
2.5 Learning with Graphs

In this section, we describe three representative applications based on our graphs, including clustering, transductive and inductive semi-supervised classification.

2.5.1 Graph based Clustering

We extend a classical spectral clustering method, normalized cut (NCut) [66], on the basis of our graph. According to Section 2.3 and Section 2.4, we can learn binary graphs \( S \) and low-rank representation coefficients \( Z \). As we discussed above, \( |(Z^\top Z)_{i,j}| \) is a good choice for measuring the similarity between a pair of samples \( x_i \) and \( x_j \). Thus, we can also re-weight the graph \( S \) using it, and finally obtain a non-negative, sparse, symmetric, weighted and balanced graph \( W \), where \( W_{i,j} = S_{i,j} \circ |(Z^\top Z)_{i,j}|. \) In Model-I, \( S \) is unbalanced and thus \( W \) is also unbalanced, but \( S \) and \( W \) obtained by Model-II are balanced.

Let \( L \) denote the Laplacian matrix, \( L = D - W \), and \( D \) is the diagonal matrix whose diagonal entry \( D_{i,i} \) is the sum of the entries of column \( i \) in matrix \( W \). The normalized cut criterion can be formulated as the following trace maximization problem [66]:

\[
\arg \max_Y \frac{1}{K} \text{trace}(Y^\top WY),
\]

where \( Y = R(R^\top DR)^{-1/2}, \) \( R \) is an indicator matrix for the partitions, and \( K \) is the number of clusters.

Let \( \hat{Y} = D^{1/2}Y \), this problem is rewritten as \( \arg \max \text{trace}(\hat{Y} D^{-1/2}WD^{-1/2}\hat{Y}) \). A well known solution to this problem is achieved by setting \( \hat{Y} \) to be the top \( K \) eigenvectors of the matrix \( D^{-1/2}WD^{1/2} \).

2.5.2 Transductive Semi-supervised Classification

Given a sample set \( X \) that contains a set of \( l \) labeled samples \( X_l \) and a set of \( u \) unlabeled samples \( X_u \), \( X = [x_1, \cdots, x_l, x_{l+1}, \cdots, x_{l+u}] = [X_l, X_u], n = l + u. \) The aim of graph based semi-supervised classification is to infer the missing labels of \( X_u \) with the aid of labeled sample set \( X_l \). We use the same graph re-weighting strategy as shown in Section 2.5.1, and obtain the weighted graph \( W \), where \( W_{i,j} = S_{i,j} \circ |(Z^\top Z)_{i,j}|. \)

Our graph \( W \) can be easily combined with a representative label propagation method, Gaussian harmonic function (GHF) [78]. We denote \( Y = [Y_l, Y_u] \), where \( Y_l \) contains the probability vectors for the labeled samples and \( Y_u \) for the unlabeled samples.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Algorithm 2. Transductive Semi-supervised Learning

Input: data matrix $X = [X_l, X_u] = [x_1, x_2, \ldots, x_n]$, dictionary $A = X$, parameters $\lambda_1$ and $\lambda_2$

Output: $Y_u$

1. Normalize all the samples $x_i$ to unit-norm, $x_i = x_i / \|x_i\|$.
2. Solve problems (2.4) or (2.14) according to Section 2.3 and Section 2.4, and obtain optimal solutions $S$ and $Z$.
3. Graph re-weighting: $W_{i,j} = S_{i,j} \circ |(Z^\top Z)_{i,j}|$.
4. Calculate probability vectors $Y_u$ for unlabeled samples $X_u$ using (2.18).

The predicted probability vectors can then be obtained for unlabeled samples by:

$$
Y_u = -Y_l L_{lu} L_{uu}^{-1},
$$

(2.18)

where $L$ is the Laplacian matrix.

These steps are summarized in Algorithm 2. Same as [91], we use sample matrix, $X$, as the dictionary. Besides GHF, our graph can also be combined with other label propagation schemes such as local and global consistency (LGC) [79] and linear neighborhood propagation (LNP) [80].

2.5.3 Inductive Semi-supervised Classification

We adopt the semi-supervised discriminant analysis (SDA) method [12] for inductive semi-supervised classification. SDA incorporates a graph based smoothness regularization term to extend the objective function of linear discriminant analysis (LDA). The objective function of SDA is:

$$
\arg \max_a \frac{a^\top S_B a}{a^\top (S_T + \alpha X L X^\top) a},
$$

(2.19)

where $X$ is the training sample set, $a$ is a transformation matrix, $S_B$ and $S_T$ are the between-class and total scatter matrices defined in LDA [6], $L = D - W$ is the Laplacian matrix, and $W$ is the weighted graph learned by our models.

The optimal solution $a$ that maximizes the objective function is given by the maximum eigenvalue solution to the generalized eigenvalue problem:

$$
S_B a = \lambda(S_T + \alpha X L X^\top) a.
$$

(2.20)

Finally, we can project all samples onto the learned subspace $a$, and classify the test samples using nearest neighbor classifier.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

2.6 Experiments

In this section, we first introduce four databases used to evaluate our methods, and then compare our graphs with some state-of-the-art graphs in data clustering, transductive and inductive semi-supervised learning.

2.6.1 Databases and Settings

In our experiments four image databases are used: Extended YaleB [112], PIE [113], ORL [114] and USPS [115].

Extended YaleB Face Database. This database has 38 subjects and approximately 64 images under different illuminations per subject. We use the images of the first 15 subjects, which we crop and resize to the size of $32 \times 32$ pixels.

PIE Face Database. The PIE face database consists of 41368 images of 68 subjects. Each subject is shown in different poses, illuminations and expressions. We use the first 15 subjects, five near front poses (C05, C07, C09, C27, C29), and all the different illuminations and expressions. Each image is cropped and resized to the size of $32 \times 32$ pixels.

ORL Face Database. The ORL face database contains 400 images of 40 individuals. These images were captured at different times, under varying lighting conditions and showing different facial expressions. We crop and resize each image to $32 \times 32$ pixels.

USPS Digit Database. The USPS digit database consists of 9298 handwritten digit images of 10 numbers (0-9). The size of each image is $16 \times 16$ pixels.

In the experiments, we randomly select 50 images of every class in the PIE and YaleB databases, 100 images from each class in the USPS database, and use all the images of ORL database. We implement the proposed models in Matlab, and carry out the experiments on an Intel® Xeon® 3.07GHz processor with 8GB memory. Specifically, we compare with the following graphs:

(1) \textbf{$k$-NN graph}. In this graph, two samples are connected if one is among the $k$ nearest neighbors of the other. The $k$-NN-graph is constructed under two conditions. In \textbf{$k$-NN-I}, the number of nearest neighbors is set to 5; and in \textbf{$k$-NN-II}, this number is set to 8. We use the Gaussian kernel to re-weight the edges, and the parameter $\delta$ is adjusted to achieve the best performance on different databases.

(2) \textbf{$b$-matching (BM) graph}. We follow the algorithms described in [87], and construct a weighted and balanced graph. The parameter $b$ is selected to achieve the best results.


Table 2.1: Average accuracies (with standard deviations) of different graphs for clustering on four databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YaleB</th>
<th>PIE</th>
<th>ORL</th>
<th>USPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>k-NN-I</td>
<td>51.03±0.88</td>
<td>65.25±6.14</td>
<td>64.07±2.11</td>
<td>77.32±6.90</td>
</tr>
<tr>
<td>k-NN-II</td>
<td>64.17±3.47</td>
<td>74.07±2.78</td>
<td>81.75±3.05</td>
<td>75.98±7.89</td>
</tr>
<tr>
<td>BM [87]</td>
<td>67.19±4.91</td>
<td>74.27±2.33</td>
<td>83.65±1.59</td>
<td>78.56±2.40</td>
</tr>
<tr>
<td>$l_1$-graph [96]</td>
<td>81.40±3.29</td>
<td>83.35±3.24</td>
<td>71.70±3.99</td>
<td>80.24±4.56</td>
</tr>
<tr>
<td>LRR [91]</td>
<td>74.28±5.87</td>
<td>73.41±3.47</td>
<td>77.50±2.59</td>
<td>81.08±3.00</td>
</tr>
<tr>
<td>LRCB [93]</td>
<td>82.90±2.08</td>
<td>85.25±2.71</td>
<td>85.62±2.25</td>
<td>79.71±4.04</td>
</tr>
<tr>
<td>Ours-I</td>
<td>84.13±4.17</td>
<td>85.86±1.12</td>
<td>86.22±3.22</td>
<td>80.45±3.23</td>
</tr>
<tr>
<td>Ours-II</td>
<td><strong>85.45±2.26</strong></td>
<td><strong>87.09±2.83</strong></td>
<td><strong>88.75±2.08</strong></td>
<td><strong>81.64±3.51</strong></td>
</tr>
</tbody>
</table>

(3) $l_1$ graph. We construct the $l_1$ graph according to [96], and also symmetrize this graph.

(4) **Sparse probability graph (SPG)**. Following the algorithms in [97], we construct the non-negative sparse probability graph (SPG).

(5) **LRR graph**. In accordance with [92], we construct the LRR graph and symmetrize it. We adopt the same parameters as described in [92].

(6) **NNLRS graph**. We construct the non-negative sparse and low-rank graph according to [104]. We also symmetrize this graph. The parameters are set as described in [104].

(7) **LRCB graph**. We also compare with our previous work, LRCB graph [93]. Two parameters $\lambda_1$ and $\lambda_2$ are separately set as 2 and 0.03 to obtain the best performance.

### 2.6.2 Spectral Clustering with Graph

In this subsection, we evaluate the performance of $k$-NN graph, BM graph [87], $l_1$ graph [96], LRR graph [91], LRCB graph [93], and our two models on spectral clustering.

We utilize two metrics to evaluate the clustering performance, which are accuracy (AC) and normalized mutual information (NMI). Assume that $\hat{Y}$ is the clustering result (i.e., label vector) and $\hat{Y}$ is the ground truth. The AC is defined as:

$$AC = \frac{1}{N} \sum_{j=1}^{N} \delta(\hat{Y}(j), \text{Map}_Y \hat{Y}(j)),$$

where $N$ is the total number of samples, $\delta(x, y)$ equals to 1 if and only if $x = y$. $\text{Map}_Y \hat{Y}$ denotes the best mapping function that permutes $Y$ to match $\hat{Y}$. Here we use the Hungarian algorithm to find the best matching.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

The NMI is defined as:

\[
\text{NMI}(X,Y) = \frac{\text{MI}(X,Y)}{\max(H(X),H(Y))},
\]

where \(X\) and \(Y\) are two index sets related to \(\hat{Y}\) and \(\bar{Y}\), respectively. \(\text{MI}(X,Y)\) denotes the mutual information between \(X\) and \(Y\),

\[
\text{MI}(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log_2 \left( \frac{p(x,y)}{p(x)p(y)} \right),
\]

\(p(x)\) and \(p(y)\) represent the marginal probability distribution functions of \(X\) and \(Y\), respectively. \(p(x,y)\) is the joint probability distribution function. \(H(X)\) and \(H(Y)\) denote the entropies of \(p(x)\) and \(p(y)\). We can observe that the NMI varies between 0 and 1. Moreover, NMI does not require the matching of \(X\) and \(Y\) in advance.

Fig. 2.3 illustrates that our approach converges quickly. The relative error is calculated by \(\|X - AZ - E\|_F/\|X\|_F\). In our two models, there are two major parameters, \(\lambda_1\) and \(\lambda_2\). To choose proper values for them, we adopt a coarse-to-fine strategy to tune the parameters. We first evaluate the parameter sensitivity on our Model-II. Fig. 2.4 shows the AC and NMI of our model under different settings of \(\lambda_1\) and \(\lambda_2\), respectively. Here, \(\lambda_1\) is used to handle the corruptions or large noise in the samples, while \(\lambda_2\) is used to balance low-rank approximation and sparsification constraint. Since the images in Extended YaleB database are captured in a relatively controlled environment, there’s not much corruptions and our graph is not sensitive to \(\lambda_1\) over a wide range. Fig. 2.4 also shows that our graph achieves stable performance when \(\lambda_2\) is varied from 0.03 to 0.06, and \(\lambda_1\) does not influence the results significantly. On other three databases, we obtain similar results. Thus, for all the four databases, \(\lambda_1\) and \(\lambda_2\) are set to 2 and 0.04, respectively.

In our Model-II, the parameter \(b\) is set as 20 on four databases, as a relative larger \(b\) can lead to stable structure of graph in unsupervised learning scenario. If some labeled samples are available, \(b\) is not necessary to be set to large values. In the semi-supervised classification experiments shown in the next subsection, \(b\) is set to 10.

We repeat each method 10 times. Table 2.1 and Table 2.2 show the average accuracies and average NMI (with standard deviations) of all compared graphs on four databases. For simplicity, our Model-I and Model-II are separately denoted as Ours-I and Ours-II. We can observe that our two models always achieve better performance than other compared methods. In particular, our balanced graph (Ours-II) usually outperforms the unbalanced one (Ours-I), which further demonstrates the merits of the balanced property.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Figure 2.3: Convergence curve of our approach on PIE database. ($\rho = 1.2$, $\mu = 10^{-3}$ and $\epsilon = 10^{-5}$)

Table 2.2: Average normalized mutual information (NMI) of different graphs with standard deviations for clustering on four databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YaleB</th>
<th>PIE</th>
<th>ORL</th>
<th>USPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-NN-I</td>
<td>0.5988±0.0463</td>
<td>0.3269±0.0260</td>
<td>0.8127±0.0002</td>
<td>0.6972±0.0265</td>
</tr>
<tr>
<td>$k$-NN-II</td>
<td>0.4231±0.0179</td>
<td>0.2636±0.0152</td>
<td>0.7990±0.0030</td>
<td>0.7100±0.0191</td>
</tr>
<tr>
<td>BM [87]</td>
<td>0.4516±0.0170</td>
<td>0.5127±0.0185</td>
<td>0.8032±0.0146</td>
<td>0.7020±0.0169</td>
</tr>
<tr>
<td>$l_1$-graph [96]</td>
<td>0.5216±0.0167</td>
<td>0.4958±0.0150</td>
<td>0.7814±0.00294</td>
<td>0.6272±0.0249</td>
</tr>
<tr>
<td>LRR [92]</td>
<td>0.7122±0.0078</td>
<td>0.6060±0.0311</td>
<td>0.7799±0.0259</td>
<td>0.6693±0.0048</td>
</tr>
<tr>
<td>LRCB [93]</td>
<td>0.8541±0.0104</td>
<td>0.6463±0.0078</td>
<td>0.8126±0.0125</td>
<td>0.7083±0.0155</td>
</tr>
<tr>
<td>Ours-I</td>
<td><strong>0.8716±0.01387</strong></td>
<td><strong>0.6514±0.0146</strong></td>
<td><strong>0.8424±0.0216</strong></td>
<td><strong>0.7069±0.0095</strong></td>
</tr>
<tr>
<td>Ours-II</td>
<td>0.8673±0.0166</td>
<td><strong>0.6742±0.0107</strong></td>
<td><strong>0.8751±0.0094</strong></td>
<td><strong>0.7154±0.0102</strong></td>
</tr>
</tbody>
</table>

2.6.3 Semi-supervised Classification with Graph

Transductive Setting We first normalize all the images to be unit-norm as shown in Algorithm 2. All methods are repeated 10 times, and each time we randomly select a subset of images for each individual to create a labeled sample set. Unlike most existing semi-supervised learning experiments, we test the performance of all compared methods with only a small set of labeled samples, because the goal of semi-supervised learning is to deal with practical tasks that have very limited labeled samples but a large amount of unlabeled ones. For each individual, the percentage of labeled samples on the Extend YaleB, PIE and USPS databases is varied from 5% to 30%. Since there are only 10 images of each individual in ORL database, this percentage varies from 10% to 50%. In our models, $\lambda_1$ and $\lambda_2$ are empirically set to 2 and 0.03, and $b$ is set to 10.

Table 2.3 and Table 2.4 show the average accuracies of different graphs combined with
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Figure 2.4: AC and NMI of our graph under different values of $\lambda_1$ and $\lambda_2$ on Extended YaleB face database.

Table 2.3: Average accuracies of different graphs with standard deviations combined with the GHF label propagation method under different percentages of labeled samples (shown in the parenthesis) on YaleB and PIE databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YaleB (10%)</th>
<th>YaleB (20%)</th>
<th>YaleB (30%)</th>
<th>PIE (10%)</th>
<th>PIE (20%)</th>
<th>PIE (30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-NN-I</td>
<td>65.41±1.81</td>
<td>72.10±1.68</td>
<td>75.92±1.64</td>
<td>51.69±2.69</td>
<td>62.30±1.71</td>
<td>68.99±1.68</td>
</tr>
<tr>
<td>$k$-NN-II</td>
<td>56.03±2.27</td>
<td>64.52±1.85</td>
<td>69.81±2.36</td>
<td>44.93±3.88</td>
<td>56.90±2.48</td>
<td>62.38±1.75</td>
</tr>
<tr>
<td>BM [87]</td>
<td>63.45±2.23</td>
<td>72.30±2.43</td>
<td>76.61±2.74</td>
<td>56.84±2.86</td>
<td>67.83±2.39</td>
<td>74.67±1.95</td>
</tr>
<tr>
<td>$l_1$-graph [96]</td>
<td>58.81±13.72</td>
<td>80.93±2.57</td>
<td>88.11±2.38</td>
<td>41.41±11.79</td>
<td>72.33±5.88</td>
<td>82.25±2.28</td>
</tr>
<tr>
<td>SPG [97]</td>
<td>58.90±2.18</td>
<td>72.25±2.04</td>
<td>81.22±2.18</td>
<td>57.04±3.28</td>
<td>72.83±2.69</td>
<td>80.57±1.98</td>
</tr>
<tr>
<td>LRR [92]</td>
<td>75.48±4.02</td>
<td>88.67±1.75</td>
<td>92.95±1.55</td>
<td>59.67±6.51</td>
<td>72.83±2.69</td>
<td>89.71±1.92</td>
</tr>
<tr>
<td>NNLRS [104]</td>
<td>76.89±3.34</td>
<td>89.58±1.30</td>
<td>93.20±1.49</td>
<td>64.43±5.12</td>
<td>85.17±2.75</td>
<td>90.88±1.62</td>
</tr>
<tr>
<td>LRCB [93]</td>
<td>90.67±0.93</td>
<td>91.61±0.75</td>
<td>94.02±0.99</td>
<td>84.06±2.06</td>
<td>89.72±1.91</td>
<td>91.30±1.64</td>
</tr>
<tr>
<td>Ours-I</td>
<td>91.05±1.60</td>
<td>92.19±1.70</td>
<td>94.30±1.82</td>
<td>86.25±1.27</td>
<td>89.41±2.03</td>
<td>92.02±1.43</td>
</tr>
<tr>
<td>Ours-II</td>
<td>91.56±1.05</td>
<td>91.98±2.11</td>
<td>94.83±1.47</td>
<td>87.66±1.90</td>
<td>92.70±1.81</td>
<td>93.52±1.39</td>
</tr>
</tbody>
</table>

Table 2.4: Average accuracies of different graphs with standard deviations combined with the GHF label propagation method under different percentages of labeled samples (shown in the parenthesis) on ORL and USPS databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>ORL (10%)</th>
<th>ORL (20%)</th>
<th>ORL (30%)</th>
<th>USPS (10%)</th>
<th>USPS (20%)</th>
<th>USPS (30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-NN-I</td>
<td>59.33±1.44</td>
<td>70.41±2.43</td>
<td>76.21±1.76</td>
<td>89.40±0.92</td>
<td>90.65±0.84</td>
<td>91.31±0.66</td>
</tr>
<tr>
<td>$k$-NN-II</td>
<td>48.94±2.19</td>
<td>60.69±3.43</td>
<td>67.89±2.98</td>
<td>88.88±1.47</td>
<td>90.51±0.82</td>
<td>90.90±0.72</td>
</tr>
<tr>
<td>BM [87]</td>
<td>58.33±2.01</td>
<td>72.40±1.69</td>
<td>78.79±2.55</td>
<td>88.92±0.91</td>
<td>91.30±0.92</td>
<td>91.63±0.78</td>
</tr>
<tr>
<td>$l_1$-graph [96]</td>
<td>43.06±2.74</td>
<td>66.56±3.93</td>
<td>73.36±2.25</td>
<td>34.43±7.47</td>
<td>67.65±5.54</td>
<td>77.79±3.42</td>
</tr>
<tr>
<td>SPG [97]</td>
<td>62.78±3.02</td>
<td>77.50±2.69</td>
<td>77.14±2.18</td>
<td>61.64±0.93</td>
<td>72.35±1.36</td>
<td>80.91±1.30</td>
</tr>
<tr>
<td>LRR [92]</td>
<td>60.69±2.59</td>
<td>76.78±1.91</td>
<td>83.04±2.59</td>
<td>62.09±9.91</td>
<td>83.19±1.82</td>
<td>85.91±1.54</td>
</tr>
<tr>
<td>NNLRS [104]</td>
<td>61.27±2.76</td>
<td>77.81±2.94</td>
<td>84.75±2.59</td>
<td>80.86±5.64</td>
<td>90.85±2.71</td>
<td>91.01±1.71</td>
</tr>
<tr>
<td>LRCB [93]</td>
<td>76.11±2.41</td>
<td>82.57±2.23</td>
<td>87.70±1.85</td>
<td>89.16±0.73</td>
<td>91.41±0.79</td>
<td>92.06±0.48</td>
</tr>
<tr>
<td>Ours-I</td>
<td>83.75±0.69</td>
<td>89.04±2.33</td>
<td>91.57±2.15</td>
<td>89.78±9.91</td>
<td>91.60±1.05</td>
<td>92.45±1.63</td>
</tr>
<tr>
<td>Ours-II</td>
<td>85.16±0.22</td>
<td>90.42±1.91</td>
<td>94.50±1.03</td>
<td>89.54±1.09</td>
<td>92.03±0.96</td>
<td>92.98±1.75</td>
</tr>
</tbody>
</table>
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

GHF label propagation strategy on four databases, when the percentages of labeled samples are 10%, 20% and 30%. Fig. 2.5 shows the average accuracies versus varying percentages of labeled samples. We can observe from Tables 2.3, 2.4 and Fig. 2.5 that:

1. BM graph usually outperforms $k$-NN graphs, since BM graph emphasises sparsity and balance at the same time. But they all use traditional similarity metrics. In addition, a large number of labeled samples with small variance allow k-NN graphs to obtain impressive performance on the USPS database.

2. The advanced sparse representation based graph, SPG, outperforms $k$-NN graph in many cases on the Extended YaleB and PIE databases, and achieves comparable performance as BM;

3. NNLRS graph, which is a low-rank and sparse graph, performs better than other graphs in most cases;

4. When the percentage of labeled samples is increased, the performance of all compared methods is increased. Our two graphs outperform other compared graphs in almost all cases, and it reaches great performance very quickly. When the labeled samples are very limited, e.g., under 10% of total number of samples, our graphs can significantly improve the accuracy over the state-of-the-art graphs on three face databases.

5. Results show that Ours-II achieves better results than Ours-I in most cases, which demonstrates the effectiveness of balanced property.

Inductive Setting Our graphs can be also concatenated with inductive semi-supervised learning algorithms, such as semi-supervised discriminant analysis (SDA) [12]. On the YaleB, PIE and USPS databases, we randomly select 20 images as training samples, and the remaining images as test samples. The percentage of labeled samples varies from 5% to 30%. On the ORL database, 5 images from each subject are randomly selected to construct the training set, and the remaining images are used for testing. The number of labeled samples varies from 1 to 5. Each method is repeated 10 times. Table 2.5, Table 2.6 and Fig. 2.6 show the average accuracies of different graphs on four databases. We can observe that our two graphs obtain better performance than other graphs. Even though Ours-I graph is unbalanced, it performs better than our previous work LRCB graph that is balanced. The reason is that Ours-I, as well as Ours-II, reformulates the rank-minimization model to obtain a new similarity metric, which is the key during graph construction.
Figure 2.5: Accuracy of transductive semi-supervised classification of our models and compared methods versus different percentages of labeled samples on four databases.

Table 2.5: Average accuracies of different graphs with standard deviations combined with the semi-supervised discriminant analysis (SDA) method under different percentages of labeled samples (shown in the parenthesis) on YaleB and PIE databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YaleB (10%)</th>
<th>YaleB (20%)</th>
<th>YaleB (30%)</th>
<th>PIE (10%)</th>
<th>PIE (20%)</th>
<th>PIE (30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-NN-I</td>
<td>57.87±4.99</td>
<td>76.11±1.80</td>
<td>83.09±2.01</td>
<td>64.22±1.73</td>
<td>80.42±2.13</td>
<td>86.16±3.17</td>
</tr>
<tr>
<td>$k$-NN-II</td>
<td>52.87±3.84</td>
<td>71.18±1.87</td>
<td>79.93±2.09</td>
<td>60.31±2.01</td>
<td>77.29±2.11</td>
<td>83.89±2.50</td>
</tr>
<tr>
<td>BM [87]</td>
<td>64.96±4.54</td>
<td>80.76±1.54</td>
<td>86.33±2.53</td>
<td>67.00±3.07</td>
<td>83.04±1.98</td>
<td>87.78±1.80</td>
</tr>
<tr>
<td>$l_1$-graph [96]</td>
<td>70.42±3.64</td>
<td>80.71±2.05</td>
<td>86.04±1.85</td>
<td>73.76±2.67</td>
<td>84.24±2.02</td>
<td>88.04±2.11</td>
</tr>
<tr>
<td>LRR [92]</td>
<td>70.02±3.72</td>
<td>78.89±2.45</td>
<td>84.53±2.56</td>
<td>70.20±3.69</td>
<td>81.82±2.02</td>
<td>87.29±1.77</td>
</tr>
<tr>
<td>NNLRS [104]</td>
<td>71.54±2.52</td>
<td>80.43±1.97</td>
<td>84.79±2.19</td>
<td>72.73±3.47</td>
<td>83.11±2.64</td>
<td>87.60±1.49</td>
</tr>
<tr>
<td>LRCB [93]</td>
<td>73.05±2.01</td>
<td>80.97±1.66</td>
<td>85.22±2.13</td>
<td>76.47±2.39</td>
<td>85.05±1.99</td>
<td>88.35±2.43</td>
</tr>
<tr>
<td>Ours-I</td>
<td>75.58±3.02</td>
<td>82.67±1.85</td>
<td>86.62±1.57</td>
<td>78.36±1.61</td>
<td>85.44±2.48</td>
<td>88.29±2.59</td>
</tr>
<tr>
<td>Ours-II</td>
<td>73.67±2.15</td>
<td><strong>87.34±1.51</strong></td>
<td><strong>89.33±1.69</strong></td>
<td>76.55±2.52</td>
<td><strong>87.95±1.70</strong></td>
<td><strong>90.29±2.59</strong></td>
</tr>
</tbody>
</table>
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Table 2.6: Average accuracies of different graphs with standard deviations combined with the semi-supervised discriminant analysis (SDA) method under different percentages of labeled samples (shown in the parenthesis) on ORL and USPS databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>ORL (20%)</th>
<th>ORL (60%)</th>
<th>ORL (100%)</th>
<th>USPS (10%)</th>
<th>USPS (20%)</th>
<th>USPS (30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$-NN-I</td>
<td>83.70±2.83</td>
<td>91.40±2.82</td>
<td>94.65±1.79</td>
<td>69.55±2.92</td>
<td>75.51±2.16</td>
<td>78.15±2.65</td>
</tr>
<tr>
<td>$k$-NN-II</td>
<td>83.10±3.25</td>
<td>90.90±2.85</td>
<td>94.00±1.86</td>
<td>68.31±2.01</td>
<td>77.29±2.11</td>
<td>80.89±2.50</td>
</tr>
<tr>
<td>BM [87]</td>
<td>82.50±1.42</td>
<td>92.00±1.75</td>
<td>94.50±1.53</td>
<td>67.00±3.07</td>
<td>75.08±1.98</td>
<td>80.78±1.80</td>
</tr>
<tr>
<td>$l_1$-graph [96]</td>
<td>84.10±3.15</td>
<td>92.00±2.33</td>
<td>94.95±1.44</td>
<td>55.10±3.67</td>
<td>67.40±3.09</td>
<td>73.35±1.25</td>
</tr>
<tr>
<td>LRR [92]</td>
<td>83.20±2.73</td>
<td>91.55±2.54</td>
<td>94.85±1.55</td>
<td>57.20±4.04</td>
<td>68.70±3.84</td>
<td>74.19±1.57</td>
</tr>
<tr>
<td>NNLRS [104]</td>
<td>83.75±1.52</td>
<td>91.90±2.63</td>
<td>94.50±1.65</td>
<td>62.50±3.47</td>
<td>70.82±2.05</td>
<td>76.33±1.64</td>
</tr>
<tr>
<td>LRCB [93]</td>
<td>84.25±1.86</td>
<td>92.10±1.49</td>
<td>94.65±1.52</td>
<td>70.15±1.68</td>
<td>75.53±2.08</td>
<td>77.61±3.11</td>
</tr>
<tr>
<td>Ours-I</td>
<td>85.25±3.06</td>
<td>92.00±1.56</td>
<td>94.95±1.75</td>
<td>70.63±2.66</td>
<td>76.21±2.87</td>
<td>78.19±1.60</td>
</tr>
<tr>
<td>Ours-II</td>
<td>83.75±3.06</td>
<td>92.20±2.32</td>
<td>94.80±1.42</td>
<td>70.04±2.87</td>
<td>78.54±2.59</td>
<td>79.60±1.78</td>
</tr>
</tbody>
</table>

Figure 2.6: Accuracy of inductive semi-supervised classification of our approach and compared methods versus different percentages of labeled samples on four databases.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

2.6.4 Discussions

Our approach is expected to work well when the data have clear subspace structures, as it estimates the graph weights by taking advantage of the subspace structure of samples. As we know, face images of different person usually lie on separate subspaces. Therefore, our approach achieved much better results than other baselines on three face databases (ORL, Extended YaleB and PIE). However, the subspace structures in digit database are not very clear, as some digits are quite similar, such as 1 and 7, 0 and 6. Thus, the results of our approach and other baselines are close. Sometimes, \( k \)-NN can get better results than other methods.

Another interesting phenomenon is that, in semi-supervised classification, our method works very well even if the labeled samples are very limited. In this case, the similarity metrics of some baselines (e.g., Gaussian kernel in \( k \)-NN graph) are not robust to noise, and therefore the estimated graph weights are unreliable. In our approach, we explicitly model the noise contained in samples, and calculate graph weights using low-rank codings. Experiments on several face databases showed that, although the face images contain illumination or pose changes, our approach still obtains impressive results with a few labeled samples.

To illustrate why our graph outperforms other compared graphs, Fig. 2.7 visualizes several weighted graphs including \( k \)-NN graph, \( l_1 \)-graph, LRR graph, and our two graphs on the PIE face database. We can observe from Fig. 2.7 that \( k \)-NN graph is sparse, but it’s not balanced. \( l_1 \)-graph is not as sparse as \( k \)-NN. LRR produces a very dense graph that is undesirable for graph based learning algorithms. Ours-I graph in Fig. 2.7(d) can recover the block diagonal structures clearly, and the similarity values in diagonal blocks are much higher than those of \( l_1 \) and LRR graphs, but it’s still not balanced. Fig. 2.7(e) shows that Ours-II graph is much sparser than the aforementioned graphs. It correctly connects the samples within each class (diagonal blocks in the graph), and meanwhile the misconnections between samples in different classes are fewer than other graphs. The results validate the superiority of our low-rank coding based similarity metric, as well as the balanced property.

Another consideration is the computational cost of our approach. Although the MM-ALM algorithm presented in Algorithm 1 is more efficient than other solvers, the \( b \)-matching optimization has a high computational cost. Table 2.7 shows the average computational time of different graph construction methods on the PIE database. KNN graph is the most efficient one, and \( l_1 \) and LRR graph have similar time costs. Ours-II consumes the most time because of the \( b \)-matching constraint. As we can see, Ours-I model offers a good balance between accuracy and efficiency.
CHAPTER 2. ROBUST GRAPH CONSTRUCTION

Figure 2.7: Visualization of different weighted graphs on PIE face database. (Please enlarge the figure to see details in graphs)

Table 2.7: Average time cost (seconds) on PIE database.

<table>
<thead>
<tr>
<th>Method</th>
<th>$k$-NN</th>
<th>$l_1$ [96]</th>
<th>LRR [92]</th>
<th>Ours-I</th>
<th>Ours-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>1.5</td>
<td>144.8</td>
<td>205.9</td>
<td>365.8</td>
<td>516.3</td>
</tr>
</tbody>
</table>

2.7 Summary

In this chapter, we have proposed a novel graph construction approach for graph based learning, including data clustering and semi-supervised classification. By taking advantages of low-rank coding and sparsification constraints (i.e., $k$-NN and $b$-matching), we jointly learned symmetric and sparse graphs. We also designed novel optimization algorithms to solve the proposed models. Experimental results on the Extended YaleB, PIE, ORL and USPS databases demonstrated the effectiveness of our approach compared with several state-of-the-art methods.
Chapter 3

Robust Subspace Discovery

3.1 Background and Motivation

Subspace learning methods have been extensively studied in pattern recognition and data mining areas during the last two decades [116, 6, 117, 118, 119, 120, 121]. Some representative subspace learning methods include principal component analysis (PCA) [116], linear discriminant analysis (LDA) [6], locality preserving projections (LPP) [117], neighborhood preserving embedding (NPE) [118], locality sensitive discriminant analysis (LSDA) [9], discriminative locality alignment (DLA) [122]. The basic idea of subspace learning methods is to find a low-dimensional projection which satisfies some specific properties [123]. As unsupervised methods, PCA [116] seeks such a subspace where the variance of projected samples is maximized, while LPP [117] and NPE [118] aim to find subspaces which can preserve the locality relationships of samples. When class labels are available, supervised subspace methods are more effective for classification tasks. LDA [6] aims at finding a projection which maximizes the inter-class scatter and minimizes the intra-class scatter at the same time. It extracts discriminative features for classification tasks. LSDA [9] preserves both discriminant and local geometrical structure in data. DLA [122] is designed based on the patch alignment framework which presents the idea of part optimization and whole alignment. As a discriminative model, it is suitable for the nonlinear classification problem. In [124], two generic frameworks are presented to implement supervised subspace learning for multi-label classification. Note that the frameworks built in [122, 124] provide us with unified interpretations of many subspace learning methods. LPP [117] and NPE [118] can also be extended to supervised versions. Those methods usually obtain promising results on clean data, however, when the data are corrupted by considerable noise (e.g., missing pixels or outliers) or large variations (e.g., pose variations in face...
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

images) in real applications, their performance is heavily degraded [29].

To learn effective features from noisy data, many techniques have been introduced, and sparse representation (SR) is among the most successful ones. SR has proven to be robust to noise, and has shown impressive results for face recognition under noisy conditions [95, 125]. The idea of SR has also been considered in dimensionality reduction and subspace learning [126, 127, 128, 129]. [127] combines dimensionality reduction and a sparse representation classifier (SRC). A sparsity preserving projections (SPP) method is proposed in [126], and its improved version is introduced in [129]. Moreover, a linear subspace learning (LSL) algorithm via sparse coding is described in [128], which also involves dictionary learning. Most SR methods seek the sparsest coding vector to represent each test sample by all training samples. However, the underlying global structure of data is not considered in these methods, and therefore they may not be robust to noise when extra clean data is not available [24].

Low-rank modeling has attracted a lot of attention recently, which can recover the underlying structure of data [23, 130]. It’s an extension of sparse representation. When data are drawn from a single subspace, robust PCA [23] is able to recover the corrupted data by minimizing the rank of data matrix. As an extension of RPCA, low-rank representation (LRR) [24] can recover corrupted data drawn from multiple subspaces. RPCA has been successfully applied to background modeling, and LRR achieves impressive performance on subspace clustering. Many improved versions of LRR have been developed. Latent LRR (LatLRR) [29] considers the effects of hidden data. Low-rank coding based balanced (LRCB) graph is designed for clustering [35] and semi-supervised classification [93]. In addition, low-rank modeling has been applied to outlier detection [37], domain adaption [131], transfer learning [132, 133], and dictionary learning [38, 32, 134]. Low-rank modeling usually suffers large computational burden, and the idea of divide-and-conquer has been introduced to solve this problem [135, 31], which makes low-rank modeling scalable to larger datasets.

As discussed above, low-rank modeling has shown impressive performance in various applications [23, 24, 136]. However, only a few of those methods can take advantages of class label information during low-rank learning, which is key for classification purpose. On the other hand, although the conventional subspace learning approaches usually obtain good performance for classification tasks, they have strong assumptions on the data distribution, and therefore they are sensitive to the noisy data. The learned subspace has limited discriminability. Can we leverage the advantages of both supervised subspace learning and low-rank modeling for classification?

In this chapter, we propose to exploit a discriminative and robust subspace, which is insensitive to noise or pose/illumination variations, for dimensionality reduction and classification.
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

Figure 3.1: Framework of the proposed approach. We jointly remove noise from data $X$ and learn robust subspace $P$. The corrupted samples are mixed in the original space, but they are well separated in the learned subspace.

In particular, we propose a novel linear subspace approach named Supervised Regularization based Robust Subspace (SRRS) for pattern classification. As illustrated in Figure 3.1, the core idea of our approach is to jointly learn low-rank representations from the noisy data, and a discriminative subspace from the recovered clean data. Moreover, to improve the classification performance of our approach, we naturally incorporate class label information into our objective function as supervised regularization. This regularization term enables us to learn a discriminative subspace, which benefits classification tasks. Finally, we formulate our model as a constrained rank minimization problem, and solve it using the recently proposed ALM algorithm [109]. The convexity of supervised regularization term is proved theoretically. Experimental results on six benchmark datasets show that our SRRS approach outperforms the traditional subspace methods and several state-of-the-art low-rank modeling methods in almost all cases, especially when the data contain considerable variations or are corrupted by noise.

Our work is closely related to some existing low-rank learning methods. The latent LRR (LatLRR) [29] approach could be integrated with some subspace learning methods. But the representation learnt by LatLRR does not necessarily guarantee an optimal input for the subsequent subspace learning. While our approach simultaneously seeks optimal low-rank representations and discriminative subspaces. In [39], a low-rank method with structural incoherence is applied to face
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

recognition. It first decomposes raw images into low-rank part and sparse part, and then applies PCA on the low-rank part to obtain a subspace. It does not, however, learn the low-rank representation and a discriminative subspace simultaneously. In this manner, the low-rank part is expected to be discriminative and benefit classification tasks. In [32], a structured low-rank representation method is presented for image classification. The differences between [32] and our approach include: (1) It learns a dictionary $D$ to represent the sample set $X$ in the original sample space, but our approach aims at learning a low-dimensional discriminative subspace to reduce the dimensionality of samples. (2) It enforces a diagonal structure prior on the coefficient matrix $Z$ to introduce the supervised information, but our approach employs the Fisher criterion to learn discriminative features; (3) It uses the ridge regression model for classifying new samples, but our approach adopts the nearest neighbor classifier. In [40], a LRR-based discriminative projection method (LRR-DP) is proposed for feature extraction. In this method, LRR is regarded as a data pre-processing method, and is performed only once to decompose sample set into two parts, the low-rank denoised samples and associated sparse noise. However, this decomposition is not guaranteed to be optimal for classification, as it doesn’t make use of any class prior information. On the contrary, our approach iteratively learns subspace and decomposes sample set, and it takes full advantage of class information through supervised regularization. In [41], a discriminant regularization term is incorporated into the formulation of Robust PCA. This method differs from our approach in two aspects. First, Robust PCA used in [41] can only model one single subspace, while our approach is able to discover multiple subspaces by virtue of LRR, which fits well for multi-class classification problems. Second, the method in [41] separately learns low-rank data representation and subspace, which means the obtained subspace cannot be guaranteed to be optimal, while our approach iteratively learns low-rank representations and discriminative subspaces.

The most relevant method in the literature is low-rank transfer subspace learning (LTSL) [132, 133], which incorporates low-rank constraint in subspace learning. However, there are significant differences between LTSL and our approach. First, LTSL is a transfer learning method that seeks a common subspace for two domains, while our approach lies in supervised learning. Second, LTSL employs low-rank constraint in low-dimensional subspace in order to transfer knowledge across two domains. In our approach, the low-rank constraint is enforced in the high-dimensional feature space in order to preserve more information.

In summary, our contributions include:

- We have proposed a new feature extraction framework, which smoothly integrates linear
subspace learning and low-rank matrix recovery. Supervised regularization is incorporated to improve the classification performance.

- We have designed an optimization algorithm to solve the proposed model, and have proven the convexity of the supervised regularization term.
- Besides objection recognition and face recognition evaluated in [34], we have also extended the applications of our model to kinship verification. We have also provided a more comprehensive overview of related works.

3.2 Supervised Regularization based Robust Subspace (SRRS)

In this section, a supervised regularization based robust subspace (SRRS) approach is proposed. We first formulate our approach as a regularized rank-minimization problem. To solve this problem, we develop an efficient optimization algorithm. Theoretical analysis on convexity is also provided.

3.2.1 Problem Formulation

Let $X$ denote the sample set that consists of $n$ training samples from $c$ classes, i.e., $X = [x_1, x_2, \ldots, x_n]$. Given a complete basis matrix $A = [a_1, a_2, \ldots, a_m] \in \mathbb{R}^{d \times m}$, we can represent each sample $x_i$ as a linear combination of the basis, which is

$$X = AZ,$$  \hspace{1cm} (3.1)

where $Z \in \mathbb{R}^{m \times n}$ is the coefficient matrix. As suggested in existing subspace clustering methods, $A$ is usually set as the sample set $X$, i.e., $A = X$. We will discuss the choice of basis matrix $A$ at the end of this section.

To achieve our goal of seeking a robust subspace $P \in \mathbb{R}^{d \times p}$, we first denote the projected low-dimensional sample set as $\tilde{X} = P^TX = P^TAZ$. Then we in turn incorporate low-rank constraint and supervised regularization to learn the projection $P$.

First, due to the fact that $n$ samples belong to $c$ different classes and $n \gg c$, these samples should be drawn from $c$ different subspaces, and therefore the coefficient matrix $Z$ is expected to be low-rank. In other words, the coefficient vectors corresponding to samples from the same class should be highly correlated.
Second, since class information is crucial to classification problems, we design a supervised regularization term $f(P, Z)$ based on the idea of Fisher criterion [6], that is, $f(P, Z) = \frac{\text{Tr}(S_B(P^T AZ))}{\text{Tr}(S_W(P^T AZ))}$, where $\text{Tr}(K)$ is the trace of matrix $K$. $S_B(P^T AZ)$ and $S_W(P^T AZ)$ are the between-class and within-class scatter matrices

$$S_B(P^T AZ) = S_B(\tilde{X}) = \sum_{i=1}^{c} n_i(m_i - m)(m_i - m)^T,$$

$$S_W(P^T AZ) = S_W(\tilde{X}) = \sum_{i=1}^{c} \sum_{j=1}^{n_i} (\tilde{x}_{ij} - m_i)(\tilde{x}_{ij} - m_i)^T,$$

where $m_i$ is the mean sample of the $i$-th class in $\tilde{X}$, $m$ is the overall mean sample of $\tilde{X}$, and $\tilde{x}_{ij}$ is the $j$-th sample in the $i$-th class of $\tilde{X}$.

By using Fisher criterion, the projected samples from different classes should be far apart, while projected samples from the same class should be close to each other. Furthermore, [137] pointed out that this trace-ratio problem can be converted into a trace difference problem. We then rewrite $f(P, Z)$ as $\bar{f}(P, Z) = \text{Tr}(S_W(P^T AZ)) - \text{Tr}(S_B(P^T AZ))$.

Based on the above observations, we come up with the following objective function

$$\min_{Z, P} \text{rank}(Z) + \lambda_1 \bar{f}(P, Z),$$

s.t. $X = AZ$, (3.2)

where $\lambda_1$ is a trade-off parameter to balance the low-rank and discriminative terms.

However, the rank minimization problem in objective (3.2) is difficult to solve, since $\text{rank}(\cdot)$ is a non-convex function. Fortunately, nuclear norm is a good surrogate for the rank minimization problem [23, 138, 24], and then (3.2) becomes

$$\min_{Z, P} \|Z\|_* + \lambda_1 \bar{f}(P, Z),$$

s.t. $X = AZ$, (3.3)

where $\|Z\|_*$ is the nuclear norm of a matrix (i.e., the sum of singular values of the matrix) [139].

We also notice that the second term $\bar{f}(P, Z)$ in (3.3) is not convex to $Z$ because of the term $-\text{Tr}(S_B)$, so we add an elastic term to ensure the convexity

$$\hat{f}(P, Z) = \text{Tr}(S_W) - \text{Tr}(S_B) + \eta \|P^T AZ\|_F^2. \quad (3.4)$$

We theoretically prove the convexity of (3.4) in the next section.
Equation (3.4) can be equivalently expressed as
\[
\hat{f}(P, Z) = \| P^T AZ (I - H_b) \|_F^2 - \| P^T AZ (H_b - H_t) \|_F^2 + \eta \| P^T AZ \|_F^2,
\]
(3.5)
where \( \eta \) is a trade-off parameter, \( \| \cdot \|_F \) is the Frobenius norm, \( I \) is an identity matrix in \( \mathbb{R}^{n \times n} \), and \( H_b \) and \( H_t \) are two constant coefficient matrices. In detail, \( H_b(i,j) = \frac{1}{n_c} \) only if \( x_i \) and \( x_j \) belong to the same class, where \( n_c \) is the number of samples in each class; otherwise, \( H_b(i,j) = 0 \). \( H_t(i,j) = \frac{1}{n} \).

The supervised regularization term \( \hat{f}(P, Z) \) is convex with respect to \( Z \). We will provide theoretical analysis to prove it in Section 3.2.2.

Orthogonality in a subspace means that any two basis vectors in this subspace are orthogonal to each other, which has the advantages of compactness and reducing redundancy. To this end, an orthogonal constraint \( P^T P = I_p \) is incorporated into our framework, where \( I_p \) is an identity matrix in \( \mathbb{R}^{p \times p} \). By combining equations (3.3) and (3.5), we obtain the objective function as follows
\[
\min_{Z,P} \| Z \|_* + \lambda_1 (\| P^T AZ (I - H_b) \|_F^2 - \| P^T AZ (H_b - H_t) \|_F^2 + \eta \| P^T AZ \|_F^2),
\]
(3.6)
\[\text{s.t. } X = AZ, P^T P = I_p.\]

Note that our objective function in (3.6) is not convex with respect to \( P \), because of the orthogonal constraint \( P^T P = I_p \).

In real-world applications, as we discussed in Section I, data usually contain considerable noise. To obtain robust subspaces, we should identify noisy information in raw data, and learn reliable subspaces from the recovered noise-free data. Specifically, we adopt the \( l_{2,1} \)-norm (i.e., \( \| \cdot \|_{2,1} \)) to model the noise contained in data. \( l_{2,1} \)-norm is a valid norm as it satisfies three conditions for a norm: (1) positive scalability: \( \| \alpha E \|_{2,1} = |\alpha| \| E \|_{2,1} \), where \( \alpha \) is a real scalar; (2) triangle inequality: \( \| B + E \|_{2,1} \leq \| B \|_{2,1} + \| E \|_{2,1} \); (3) existence of a zero vector: if \( \| E \|_{2,1} = 0 \), then \( A = 0 \). As \( \| E \|_{2,1} \) encourages the columns of \( E \) to be zero, the assumption in our work is that some vectors in our data are corrupted while the others are clean. Then, we have a constraint \( X = AZ + E \), and rewrite the objective function as:
\[
\min_{Z,E,P} \| Z \|_* + \lambda_2 \| E \|_{2,1} + \lambda_1 (\| P^T AZ (I - H_b) \|_F^2 - \| P^T AZ (H_b - H_t) \|_F^2 + \eta \| P^T AZ \|_F^2),
\]
(3.7)
\[\text{s.t. } X = AZ + E, P^T P = I_p.\]

where \( \| E \|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{d} ((E)_{ij})^2} \), and \( \lambda_2 \) is a trade-off parameter.
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

We have described how to jointly learn discriminative subspace and low-rank representations. In the next section, we will introduce the optimization algorithm. Other than Fisher criterion discussed above, other types of objectives, such as locality preserving, can also be easily incorporated into our framework by reformulating the regularization term \( \hat{f}(P, Z) \).

3.2.2 Theoretical Analysis

We theoretically analyze the convexity of supervised regularization term \( \hat{f}(P, Z) \) with respect to \( Z \), which is critical to ensure that our model is solvable using ALM algorithms. In particular, to guarantee the convexity of (3.5), we provide the following theorem.

**Theorem 1.** If \( \eta > 1 \), the supervised regularization term

\[
\hat{f}(P, Z) = \|P^T A(Z(I - H_b))\|_F^2 - \|P^T AZ(H_b - H_t)\|_F^2 + \eta \|P^T A(Z(H_b - H_t))\|_F^2
\]

is convex to \( Z \) when \( P \) is fixed.

**Proof.** Let \( T = P^T A Z \), where \( P^T A \) can be regarded as constant when optimizing \( Z \). We then can convert \( \hat{f}(P, Z) \) to \( f(T) \) as follows

\[
f(T) = \|T(I - H_b)\|_F^2 - \|T(H_b - H_t)\|_F^2 + \eta \|T\|_F^2. \tag{3.8}
\]

Now we can rewrite \( T \) as a column vector, \( T = [r_1, r_2, \ldots, r_n]^T \), where \( r_i \) is the \( i \)-th row vector of \( T \). Then \( f(T) \) is equivalent to

\[
f(T) = \|\text{diag}((I - H_b)^T)T\|_2^2 - \|\text{diag}((H_b - H_t)^T)T\|_2^2 + \eta \|T\|_2^2, \tag{3.9}
\]

where \( \text{diag}(K) \) is to construct a block diagonal matrix with each block on the diagonal being matrix \( K \).

The convexity of \( f(T) \) depends on whether its Hessian matrix \( \nabla^2 f(T) \) is positive definite or not. \( \nabla^2 f(T) \) will be positive definite if matrix \( S \) is positive definite.

\[
S = (I - H_b)(I - H_b)^T - (H_b - H_t)(H_b - H_t)^T + \eta I. \tag{3.10}
\]

Note that we have the equations \( H_b H_t = H_t H_b = H_t \) and \( H_t H_t = H_t \). Then, we can obtain

\[
S = (1 + \eta)I - 2H_b + H_t. \tag{3.11}
\]

To justify that if matrix \( S \) is positive definite, we employ the following lemma.

**Lemma 1** (Weyl’s Inequality; Theorem 1 [140]) Let \( G \) denote an \( n \) by \( n \) Hermitian matrix, the ordered eigenvalues of \( G \) are \( \lambda_1(G) \geq \cdots \geq \lambda_n(G) \). If \( B, C \) are \( n \) by \( n \) Hermitian matrices, then \( \lambda_n(B) + \lambda_n(C) \leq \lambda_n(B + C) \).
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

Lemma 1 tells us the smallest eigenvalue of matrix \((B + C)\) is greater than or equal to the sum of the smallest eigenvalues of \(B\) and \(C\). In our problem, we need to make \(S\) positive definite, which means the smallest eigenvalue of \(S\) should be greater than 0. Thus, we employ Lemma 1 to evaluate the equation (3.11). The minimal eigenvalues of \(-H_b\) and \(H_t\) are \(-1\) and 0, so we should ensure:

\[
(1 + \eta) - 2 + 0 > 0. \tag{3.12}
\]

Hence, we have \(\eta > 1\) from the above equation, which could guarantee that \(f(T)\) is convex to \(T\). Recall that \(T = P^T AZ\) and \(P^T A\) is a constant. Therefore, we can further conclude that \(f(P, Z)\) is convex to \(Z\) when \(\eta > 1\) and \(P\) is fixed.

3.2.3 Optimization

To solve (3.7), we adopt the recently proposed inexact augmented Lagrange multiplier (ALM) algorithm [109]. Firstly, we add a variable \(J\) and a new constraint \(Z = J\) to relax the original problem

\[
\begin{align*}
\min_{Z, E, P, J} & \quad \|J\|_* + \lambda_2 \|E\|_{2,1} + \lambda_1 (\|P^T AZ(I - H_b)\|_F^2 \\
& - \|P^T AZ(H_b - H_t)\|_F^2 + \eta \|P^T AZ\|_F^2), \\
\text{s.t.} \quad & X = AZ + E, P^T P = I_p, Z = J.
\end{align*}
\tag{3.13}
\]

Furthermore, (3.13) can be converted to the following problem

\[
\begin{align*}
\min_{Z, E, J, P, Y, R} & \quad \|J\|_* + \lambda_2 \|E\|_{2,1} + \lambda_1 (\|P^T AZ(I - H_b)\|_F^2 \\
& - \|P^T AZ(H_b - H_t)\|_F^2 + \eta \|P^T AZ\|_F^2) \\
& + \text{Tr}(Y^T (X - AZ - E)) + \text{Tr}(R^T (Z - J)) \\
& + \frac{\mu}{2} (\|X - AZ - E\|_F^2 + \|Z - J\|_F^2), \\
\text{s.t.} \quad & P^T P = I_p.
\end{align*}
\tag{3.14}
\]

where \(\mu > 0\) is a penalty parameter, \(Y \in \mathbb{R}^{d \times n}\) and \(R \in \mathbb{R}^{m \times n}\) are Lagrange multipliers.

To solve (3.14), we alternately update the variables \(P, J, Z,\) and \(E\). First, we learn a subspace \(P\) given an initialized low-rank representation matrix \(Z\). Second, on the fixed subspace \(P\), we update the low-rank representation matrix \(J, Z\) and the noise matrix \(E\). Although the convergence of inexact ALM algorithm cannot be guaranteed when there are three or more variables, some theoretical results have been presented to ensure the convergence with mild conditions [24]. In addition, we demonstrate the convergence properties of our algorithm in the experiments.

Learn Subspace \(P\) on Fixed Low-Rank Representations

42
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

We first discuss how to optimize $P$ while fixing $Z$, $J$, and $E$. Note that $\|J\|_* + \lambda_2 \|E\|_{2,1} + \text{Tr}(Y^T(X - AZ - E)) + \text{Tr}(R^T(Z - J)) + \frac{\rho}{2} (\|X - AZ - E\|^2_F + \|Z - J\|^2_F)$ can be regarded as constant.

The objective function w.r.t. $P$ becomes

$$
P_{k+1} = \min_{P_k} \lambda_1(\|P^T_k AZ_k(I - H_b)\|^2_F - \|P^T_k AZ_k(H_b - H_t)\|^2_F + \eta \|P^T_k AZ_k\|^2_F)$$

$$\text{s.t. } P_k^T P_k = I_p. \quad (3.15)$$

For simplicity, let $Z_{wk} = AZ_k(I - H_b)$, $Z_{bk} = AZ_k(H_b - H_t)$. We derive the solution to the projection vectors in $P_k$ one by one. To obtain the $i$-th column in $P_k$ (denoted as $P_{k(i,i)}$), we rewrite (3.15) as

$$P_{k+1(i,i)} = \min_{P_{k(i,i)}} \lambda_1(\|P^T_{k(i,i)} Z_{wk}\|^2_F - \|P^T_{k(i,i)} Z_{bk}\|^2_F + \eta \|P^T_{k(i,i)} AZ_k\|^2_F + \beta_i(P_{k(i,i)} - 1)), \quad (3.16)$$

where $\beta_i$ is the corresponding Lagrange multiplier.

By setting the derivative w.r.t. $P_{k(i,i)}$ to zero, we have

$$- \lambda_1(Z_{wk} Z_{wk}^T - Z_{bk} Z_{bk}^T + \eta AZ_k Z_k^T A^T) P_{k(i,i)} = \beta_i P_{k(i,i)} \quad (3.17)$$

Therefore, $P_{k(i,i)}$ is the $i$-th eigenvector of matrix $- \lambda_1(Z_{wk} Z_{wk}^T - Z_{bk} Z_{bk}^T + \eta AZ_k Z_k^T A^T)$, corresponding to the $i$-th smallest eigenvalue.

Learn Low-Rank Representations $Z$ on Fixed Subspace

Here we show how to update $J_{k+1}$, $Z_{k+1}$ and $E_{k+1}$ when fixing $P_{k+1}$. After dropping the irrelevant terms w.r.t. $J$, (3.14) can be rewritten as

$$J_{k+1} = \min_{J_k} \frac{\|J_k\|_*}{\lambda} + \text{Tr}(R^T(Z_k - J_k)) + \frac{\mu}{2} \|Z_k - J_k\|^2_F \quad (3.18)$$

Problem (3.18) can be effectively solved using the singular value thresholding (SVT) operator [139]. SVT contains two major steps. First, we perform SVD on the matrix $S = Z_k + (R_k / \mu_k)$, and get $S = U_S \Sigma_S V_S$, where $\Sigma_S = \text{diag}(\sigma_i)_{1 \leq i \leq r}$, $\sigma_i$ are the singular values with rank $r$. Second, we can obtain the optimal solution $J_{k+1}$ by thresholding the singular values: $J_{k+1} = U_S \Omega_{(1/\mu_k)}(\Sigma_S)V_S$, where $\Omega_{(1/\mu_k)}(\Sigma_S) = \text{diag}(\{\sigma_i - (1/\mu_k)\}_+), \text{ and } t_+ \text{ means the positive part of } t$.

By ignoring terms independent of $Z$ in (3.14), we have

$$\min_{Z,Y,R} \lambda_1(\|P^T AZ(I - H_b)\|^2_F - \|P^T AZ(H_b - H_t)\|^2_F + \eta \|P^T AZ\|^2_F + \text{Tr}(Y^T(X - AZ - E))) + \text{Tr}(R^T(Z - J)) + \frac{\rho}{2} (\|X - AZ - E\|^2_F + \|Z - J\|^2_F). \quad (3.19)$$
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

By setting the derivative w.r.t. $Z$ to zero, we have

$$Z_{k+1}D/\mu_k + (A^TP_{k+1}P_{k+1}^TA)^{-1}(I + A^TA)Z_{k+1} = (A^TP_{k+1}P_{k+1}^TA)^{-1}K_{k+1},$$

(3.20)

where $D = 2\lambda_1((1 + \eta)I - 2H_b + H_t^T)$, and $K_{k+1} = J_{k+1} + A^T(X - E_k) + (A^TY_k - R_k)/\mu_k$. Problem (3.20) is a standard Sylvester equation, which can be effectively solved using existing tools [141].

Similarly, after dropping terms independent of $E$, we can rewrite (3.14) as

$$E_{k+1} = \min_{E_k} \frac{\lambda_2}{\mu_k} \|E_k\|_2 + \frac{1}{2} \|E_k - (X - AZ_{k+1} + Y_k/\mu_k)\|_F^2.$$  

(3.21)

The solution to problem (22) is presented in [24]. Specifically, let $\Psi = X - AZ_{k+1} + Y_k/\mu_k$, the $i$-th column of $E_{k+1}$ is

$$E_{k+1}(:, i) = \begin{cases} \frac{\|\Psi_i\| - \frac{\lambda_2}{\mu_k}\Psi_i}{\|\Psi_i\|}, & \text{if } \frac{\lambda_2}{\mu_k} < \|\Psi_i\|, \\ 0, & \text{otherwise}. \end{cases}$$

(3.22)

As stated in the inexact ALM algorithm, we also need to update the Lagrange multipliers $Y$ and $R$, and the parameter $\mu$ after optimizing the variables $P$, $J$, $Z$ and $E$.

3.2.4 Algorithm and Discussions

The above process is repeated until convergence. The detailed algorithm of our optimization is outlined in Algorithm 1.

After obtaining the optimal solution $P^*$ and $Z^*$, we project both training samples and test samples onto $P^*$, and then utilize nearest neighbor (NN) classifier to predict the label vector of test samples. The complete procedures of our SRRS approach are summarized in Algorithm 2.

The time complexity of our approach mainly depends on the complexity of Algorithm 1. In Algorithm 1, the most time-consuming steps are Steps 2–4. Step 2 and Step 3 cost $O(n^3)$ due to the SVD decomposition, where $n$ is the total number of samples. The matrix inverse calculation in (3.20) costs $O(n^3)$, and the state-of-the-art solution to a Sylvester equation costs $O(n^3 + m^3)$ (In our case, $m = n$). In all, the overall time complexity of our approach is $O(tn^3)$, where $t$ is the number of iterations.

Formula (4) is actually a general framework for robust subspace learning and feature extraction. In this chapter, we design a supervised regularization term $\bar{f}(P, Z)$ by virtue of Fisher criterion. Other subspace learning baselines (e.g., LPP, NPE, and LFDA) could also be extended under our framework by reformulating the regularization term $\tilde{f}(P, Z)$. 

44
Algorithm 1. Solving Problem (3.14) by Inexact ALM

**Input:** data matrix $X$, parameter $\lambda_1, \lambda_2, \eta, Z = J = 0$, 
$E_0 = 0$, $Y_0 = 0$, $R_0 = 0$, $\mu_0 = 0.1$, $\mu_{\text{max}} = 10^{10}$, 
$\rho = 1.3$, $k = 0$, $\epsilon = 10^{-8}$

**Output:** $P_k, Z_k, E_k$

1: while not converged do
2: update $P_{k+1}$ using (3.16), given others fixed
   If $k = 1$, then $Z_k = I$.
3: update $J_{k+1}$ using (3.18), given others fixed
4: update $Z_{k+1}$ using (3.20), given others fixed
5: update $E_{k+1}$ using (3.21), given others fixed
6: update the multipliers $Y_{k+1}$ and $R_{k+1}$
   $Y_{k+1} = Y_k + \mu_k (X - AZ_{k+1} - E_{k+1})$
   $R_{k+1} = R_k + \mu_k (Z_{k+1} - J_{k+1})$
7: update the parameter $\mu_{k+1}$ by
   $\mu_{k+1} = \min (\rho \mu_k, \mu_{\text{max}})$
8: check the convergence conditions
   $\|X - AZ_{k+1} - E_{k+1}\|_\infty < \epsilon$ and
   $\|Z_{k+1} - J_{k+1}\|_\infty < \epsilon$.
9: $k = k + 1$
10: end while

In Algorithms 1 and 2, sample set $X$ is utilized as dictionary (i.e., $A = X$). When the sampling is insufficient, learning an informative dictionary should enhance the classification performance, which provides another interesting direction of future work.

In the Step 3 of Algorithm 2, we project the recovered clean training images $X Z$ onto the subspace $P$. Ideally, we would also like to project the clean test images onto $P$ for classification. However, it is usually not practical to obtain clean test images in real applications. In this chapter, to show the robustness of $P$ for noisy data, we directly project noisy images onto $P$. To enhance the classification performance, one could apply some image denoising techniques before projecting noisy test data onto $P$.

### 3.3 Experiments

The performance of our SRRS approach is evaluated on six benchmark datasets, including object datasets [142, 143], face datasets [112, 144], and KinFace dataset [145]. We compare our approach with related methods on the robustness to different kinds of noise including pixel corruption.
Algorithm 2. SRRS Approach

**Input:** Training sample set $X$ with label vectors $L_X$, test sample set $Y$, low-rank coefficients $Z$

**Output:** Predicted label vector $L_Y$ for test samples.

1: Normalize each sample $x_i$ to unit-norm, $x_i = x_i / \|x_i\|$.
2: Use Algorithm 1 to solve problem (3.14) and obtain optimal solution $P^*$.
3: Project $X$ and $Y$ onto $P^*$: $\tilde{X} = P^* X Z, \tilde{Y} = P^* Y$.
4: Predict the label vector $L_Y$ of $\tilde{Y}$ by using the nearest neighbor (NN) classifier.

and large pose/illumination variations. Our code is publicly available 1.

### 3.3.1 Object Recognition with Pixel Corruption

We use two object datasets, COIL-100 [142] and ALOI [143], in this experiment. The COIL dataset contains various views of 100 objects with different lighting conditions. Each object contributes 72 images, which are captured in equally spaced views. In our experiments, the images are converted to grayscale, resized to $32 \times 32$, and then the robustness is evaluated on alternative viewpoints. We normalize the samples so that they have unit norm that is favorable for optimization. Unlike most existing subspace learning experiments, we also test the robustness of different methods to noise by adding 10% pixel corruption to the original images. Some examples of corrupted object images in COIL dataset can be found in Figure 3.9.

In the experiments, we compare the proposed approach with PCA [116], LDA [6], NPE [118], LSDA [9], RPCA[23]+LDA, support vector machine (SVM) [146], FDDL [147], Latent LRR (LatLRR) [29] and DLRD [38]. PCA and LDA are two representative unsupervised and supervised subspace learning methods, and we use them as our baseline. NPE can preserve the neighborhood structure of data, which is less sensitive to outliers than PCA. Here we compare our method with the supervised version of NPE. LSDA is a discriminant analysis method which preserves both discriminant and local geometrical structural in the data. RPCA is effective in removing noise from corrupted data. Here we incorporate it with LDA as a baseline. SVM is a popular and powerful classifier. Here we compare with the nonlinear SVM classifier with RBF kernel. FDDL is a dictionary learning method that learns a discriminative dictionary using Fisher criterion. LatLRR and DLRD

1https://github.com/smilesheng/SRRS
### Table 3.1: Average recognition rates (%) with standard deviations of all compared methods on COIL object database.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Original Images</th>
<th>10% Corrupted Images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20 objects</td>
<td>40 objects</td>
</tr>
<tr>
<td>PCA [116]</td>
<td>86.42±1.11</td>
<td>83.75±1.12</td>
</tr>
<tr>
<td>LDA [6]</td>
<td>81.83±2.03</td>
<td>77.08±1.36</td>
</tr>
<tr>
<td>NPE [118]</td>
<td>82.24±2.25</td>
<td>76.01±1.04</td>
</tr>
<tr>
<td>LSDA [9]</td>
<td>82.79±1.70</td>
<td>75.01±1.14</td>
</tr>
<tr>
<td>RPCA+LDA</td>
<td>83.26±1.52</td>
<td>78.39±1.15</td>
</tr>
<tr>
<td>SVM [146]</td>
<td>86.52±1.51</td>
<td>86.73±1.40</td>
</tr>
<tr>
<td>FDDL [147]</td>
<td>88.98±0.85</td>
<td>88.45±0.64</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>92.03±1.21</td>
<td>92.51±0.65</td>
</tr>
<tr>
<td>Ours</td>
<td>92.03±1.21</td>
<td>92.51±0.65</td>
</tr>
</tbody>
</table>
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

![Recognition Rates](image)

Figure 3.2: Recognition rates of SRRS with different values of $\lambda_1$ and $\lambda_2$ on COIL dataset.

Table 3.2: $p$-value between SRRS and other methods on the COIL object database. The asterisk * indicates that the difference between Method A and Method B is statistically significant when $p = 0.05$.

<table>
<thead>
<tr>
<th>Method A vs. B</th>
<th>Original Images</th>
<th>Corrupted Images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20 objects</td>
<td>40 objects</td>
</tr>
<tr>
<td>Ours vs. PCA [116]</td>
<td>$1.0 \times 10^{-6}$</td>
<td>$3.2 \times 10^{-8}$</td>
</tr>
<tr>
<td>Ours vs. LDA [6]</td>
<td>$1.5 \times 10^{-7}$</td>
<td>$5.5 \times 10^{-9}$</td>
</tr>
<tr>
<td>Ours vs. NPE [118]</td>
<td>$2.3 \times 10^{-8}$</td>
<td>$2.7 \times 10^{-8}$</td>
</tr>
<tr>
<td>Ours vs. LSDA [9]</td>
<td>$1.3 \times 10^{-8}$</td>
<td>$2.0 \times 10^{-8}$</td>
</tr>
<tr>
<td>Ours vs. RPCA+LDA</td>
<td>$3.4 \times 10^{-9}$</td>
<td>$3.2 \times 10^{-8}$</td>
</tr>
<tr>
<td>Ours vs. SVM [146]</td>
<td>$3.1 \times 10^{-5}$</td>
<td>$5.8 \times 10^{-5}$</td>
</tr>
<tr>
<td>Ours vs. FDDL [147]</td>
<td>$2.7 \times 10^{-6}$</td>
<td>$4.6 \times 10^{-7}$</td>
</tr>
<tr>
<td>Ours vs. LatLRR [29]</td>
<td>$3.5 \times 10^{-5}$</td>
<td>$2.1 \times 10^{-5}$</td>
</tr>
<tr>
<td>Ours vs. DLRD [38]</td>
<td>$0.0279^*$</td>
<td>$7.0 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

are two low-rank modeling methods. LatLRR can effectively extract salient features for image recognition, while DLRD learns a low-rank dictionary for face recognition. Both of them also demonstrate stable performance under noisy conditions.

We randomly select 10 images per object to construct the training set, and the test set contains the rest of the images. This random selection process is repeated 20 times, and we report the average recognition rates for each compared method. In addition, we performed scalability evaluations, by increasing the number of objects from 20 to 100. For our approach and each compared method, the parameters are tuned to achieve their best performance via 5-fold cross-validation. Figure 3.2 shows the performance of SRRS with different values of $\lambda_1$ and $\lambda_2$ when the number of classes is 20. We can also observe that the performance is not very sensitive to the settings of $\lambda_2$. Further, SRRS obtains its best performance when $\lambda_1 = 10^{-1}$. It also shows that,
Figure 3.3: Recognition rates of our approach and compared subspace methods versus varying feature dimensions on the original ((a)-(d)) and corrupted ((e)-(h)) COIL object database. Note that LDA and our approach obtains at most $c - 1$ features, where $c$ is the number of classes.
SRRS achieves the best performance on the original data and corrupted data when $\lambda_2 = 10$ and $\lambda_2 = 1$, respectively.

Figure 3.3 shows the recognition rates of our approach and the compared subspace methods (PCA, LDA, NPE and LSDA) versus varying feature dimensions. It shows that our SRRS approach outperforms subspace methods in almost all cases. When the images contain noise, the recognition rates of compared subspace methods are severely degraded, but our approach can still obtain good results. Namely, the subspace derived from our approach is robust to pixel corruption. Table 3.1 shows the average recognition rates with standard deviations of all compared methods. It can be observed from Table 3.1 that the recognition rates of our approach vary slightly when the number of classes increases from 20 to 100.

The total average results are also summarized in Table 3.1. We can see that our approach and LatLRR have lower deviations than other methods, which demonstrates good scalability. When the images are corrupted, all traditional subspace methods have difficulty obtaining reasonable results. However, three low-rank modeling based methods achieve remarkable performance. In most cases, our SRRS approach achieves the best recognition results. Moreover, we utilize other levels of corruption such as 20%, 30%, 40% and 50% on COIL-20 database, and report the results in Figure 3.4. It shows that our SRRS approach consistently outperforms other methods.
Figure 3.5: Properties of our approach on ALOI dataset. (a) Convergence curve ($\rho = 1.3$, $\mu = 0.1$ and $\epsilon = 10^{-8}$). (b) Recognition rates of SRRS with different values of $\eta$.

We also performed a significance test, McNemar’s test, for the results shown in Table 3.1, in order to demonstrate the statistical significance of our approach compared with several of the most representative state-of-the-art methods. We use a significance level of 0.05. In another word, the performance difference between two methods is statistically significant, if the estimated $p$-value is lower than 0.05. Table 3.2 shows the $p$-values of comparing SRRS with other methods. From this table, the following conclusions can be reached: (1) The performance differences between our approach and the methods (PCA, LDA, NPE, LSDA, RPCA+LDA, SVM and FDDL) are statistically significant in all cases. (2) On the original dataset, the performance differences between our approach and DLRD/LatLRR are statistically significant. (3) On the corrupted dataset, the performance differences between our approach and DLRD/LatLRR are not statistically significant. The reason is that DLRD and LatLRR are also able to handle the noisy data. But our approach achieves higher recognition rates than them.

The ALOI dataset contains 1000 general object categories taken at different viewing angles. There are 72 equally spaced views in each category. In our experiments, we select the first 300 objects from this dataset. All the images are converted to gray-scale and resized to the size of $36 \times 48$. We also add 10% pixel corruption on the original images to evaluate the performance of different methods. Some examples of corrupted images in the ALOI dataset can be found in Figure 3.9.

Ten images of each object are randomly selected as training samples, and the others as test samples. This random selection process was repeated 20 times. Figure 3.5(a) shows the convergence
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

Table 3.3: Average recognition rates (%) on ALOI object database.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Original images</th>
<th>10% Corruption</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [116]</td>
<td>84.10±0.51</td>
<td>22.99±0.29</td>
</tr>
<tr>
<td>LDA [6]</td>
<td>83.46±0.38</td>
<td>23.97±0.37</td>
</tr>
<tr>
<td>NPE [118]</td>
<td>84.50±0.43</td>
<td>24.83±0.25</td>
</tr>
<tr>
<td>LSDA [9]</td>
<td>83.96±0.35</td>
<td>24.05±0.31</td>
</tr>
<tr>
<td>RPCA+LDA</td>
<td>84.62±0.48</td>
<td>27.58±0.41</td>
</tr>
<tr>
<td>SVM [146]</td>
<td>84.61±0.45</td>
<td>30.29±0.33</td>
</tr>
<tr>
<td>FDDL [147]</td>
<td>84.77±0.72</td>
<td>23.03±0.42</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>84.97±0.53</td>
<td>59.35±0.48</td>
</tr>
<tr>
<td>DLRD [38]</td>
<td>85.53±0.55</td>
<td>58.66±0.39</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>87.91±0.34</strong></td>
<td><strong>64.62±0.32</strong></td>
</tr>
</tbody>
</table>

curves of our approach on the original data and corrupted data. It shows that the relative error on corrupted data (10% percentage noise) is larger than that on the original data. But, in both cases, our approach converges very well after ten iterations. The relative error is calculated by \( \frac{\|X - AZ - E\|_F}{\|X\|_F} \). Figure 3.5(b) shows the recognition rates of SRRS when parameter \( \eta \) is selected from the range [0,3]. We observe that SRRS is not very sensitive to the choice of \( \eta \) when \( \eta > 1 \). Further, we set \( \eta \) to 1.5 to achieve the best recognition performance. Table 3.3 shows the average recognition rates with standard deviations for each compared method. It shows that SVM, FDDL and two low-rank methods obtain better performance than traditional subspace methods, and our approach outperforms all these methods on the original dataset and the corrupted dataset. In addition, by comparing Table 3.1 and Table 3.3, we can observe that, for the dataset with a large number of classes, the classification task becomes more difficult when data are corrupted.

3.3.2 Face Recognition with Illumination and Pose Variation

We also evaluate our approach on the Extended YaleB [112] and the FERET [144] face databases. The YaleB face dataset consists of 2414 frontal face images of 38 classes, and each of them contains about 64 images. Figure 3.6(a) shows the examples from the YaleB dataset. We crop and resize the images to the size of \( 28 \times 32 \), and normalize the pixel values to [0, 1].

As suggested in [29], we randomly select 30 images per class to construct the training set, and test set contains the rest of the images. This random selection procedure is repeated 20 times, and we show the average recognition rates in Table 3.4. It can be observed that supervised methods perform much better than the unsupervised method PCA. The reason is that PCA has a
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

![Sample images in YaleB, FERET and KinFace datasets.](image)

(Figure 3.6: Sample images in YaleB, FERET and KinFace datasets.)

Table 3.4: Average recognition rates (%) of all compared methods on YaleB and FERET face databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>YaleB</th>
<th>FERET</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [116]</td>
<td>72.57±0.58</td>
<td>84.00±2.11</td>
</tr>
<tr>
<td>LDA [6]</td>
<td>89.09±0.91</td>
<td>77.63±2.22</td>
</tr>
<tr>
<td>NPE [118]</td>
<td>86.01±1.37</td>
<td>71.67±1.95</td>
</tr>
<tr>
<td>LDSA [9]</td>
<td>92.94±0.88</td>
<td>73.27±3.01</td>
</tr>
<tr>
<td>RPCA+LDA</td>
<td>91.29±1.16</td>
<td>79.03±2.63</td>
</tr>
<tr>
<td>SVM [146]</td>
<td>94.93±0.75</td>
<td>88.03±2.04</td>
</tr>
<tr>
<td>FDDL [147]</td>
<td>95.10±1.31</td>
<td>86.00±2.51</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>88.76±1.26</td>
<td>84.27±2.19</td>
</tr>
<tr>
<td>DLRD [38]</td>
<td>93.56±1.25</td>
<td>83.33±2.40</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>97.75±0.58</strong></td>
<td><strong>89.84±2.01</strong></td>
</tr>
</tbody>
</table>

To evaluate the robustness to noise of the different methods, we randomly choose a percentage (from 10% to 50%) pixels and replace their values by random numbers that are uniformly distributed on [0, 1]. Figure 3.7 shows that, in noisy scenarios, low-rank modeling based methods (LatLRR, DLRD and our approach) consistently obtain better performance than other methods. Specifically, our SRRS approach can get the best performance.
Figure 3.7: Average recognition rates of all compared methods on YaleB database with different level of noise.

The FERET database contains 2,200 face images collected from 200 subjects, and each subject has 11 images. These images were captured under various poses and expressions. In this experiment, we randomly select the images from 50 individuals. Figure 3.6(b) provides images of one individual that show large pose variations. The original size of each image is 384×256. We cropped and resized them to the size of 30×25.

We randomly select 5 images of each individual as training samples, and the remaining samples are regarded as test samples. Table 3.4 lists the average recognition rates of all compared methods over 20 runs. It reflects that our approach can improve the recognition results over existing methods. Interestingly, PCA can outperform some supervised subspace methods on this database. A likely reason for this is that large pose changes of one individual produce large intra-class variations, which highly influence the performance of supervised methods.

### 3.3.3 Face Recognition with Occlusions

The AR face database contains over 4,000 facial images collected from 126 subjects. For each subject, there are 26 frontal face images, taken under different illuminations, expressions, and facial occlusions in two separate sessions. In our experiments, we strictly follow the experimental settings in [32], and conducted the following three experiments.
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

Table 3.5: Recognition rates (%) of all compared methods on AR face databases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Sunglasses</th>
<th>Scarf</th>
<th>Sunglasses+Scarf</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [116]</td>
<td>51.75</td>
<td>48.83</td>
<td>42.00</td>
</tr>
<tr>
<td>LDA [6]</td>
<td>82.25</td>
<td>81.75</td>
<td>79.82</td>
</tr>
<tr>
<td>NPE [118]</td>
<td>83.75</td>
<td>82.58</td>
<td>80.35</td>
</tr>
<tr>
<td>LSDA [9]</td>
<td>83.66</td>
<td>81.83</td>
<td>79.12</td>
</tr>
<tr>
<td>SVM [146]</td>
<td>78.37</td>
<td>65.33</td>
<td>67.71</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>76.52</td>
<td>75.24</td>
<td>76.11</td>
</tr>
<tr>
<td>DLRD [38]</td>
<td>85.26</td>
<td>83.01</td>
<td>81.56</td>
</tr>
<tr>
<td>LRDL [32]</td>
<td><strong>87.21</strong></td>
<td>83.96</td>
<td>82.15</td>
</tr>
<tr>
<td>Ours</td>
<td>85.84</td>
<td><strong>86.98</strong></td>
<td><strong>86.23</strong></td>
</tr>
</tbody>
</table>

**Sunglasses**: Some face images contain the occlusion of sunglasses, which are considered as corrupted samples. To construct the training set, we choose seven neutral images and one randomly selected image with sunglasses from each subject (session 1). The test set contains the remaining neutral images (session 2) and the rest of the images with sunglasses (2 images from session 1 and 3 images from session 2). Thus, for each individual, there are 8 training images and 12 test images. The sunglasses cover about 20% of the face image.

**Scarf**: We utilize the corrupted training images due to the occlusion of scarf. Using a similar training/test setting as above, we have 8 training images and 12 test images for each individual. The scarf covers about 40% of the face image.

**Sunglasses+Scarf**: Moreover, we consider the case where images contain both sunglasses and scarf. We select all the 7 neutral images and two corrupted images (one with sunglasses and the other with scarf) at session 1 for training. For each individual, there are 17 test images in total.

Table 3.5 shows the recognition rates of compared methods in three different scenarios. We can observe that LRDL obtains the best result in the Sunglasses case, and our approach obtains the best results in the Scarf case and the mixed case. Figure 3.9 (d) shows that our approach can correctly recover the clean images from the occluded face images. Therefore, we can train robust classifiers from the recovered image set $AZ$.

3.3.4 Kinship Verification

Kinship verification is a recently investigated research topic, which aims at determining kin relationships from photos. It is still a very challenging task due to large variations in different human faces. We also evaluate the performance of our approach and related methods on kinship
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

![Graphs showing low-rank representation coefficients for different pairs of samples.](image)

Figure 3.8: Visualization of low-rank representation coefficients of two pairs of samples on FERET database.

verification. We conduct the kinship verification experiments on the UB KinFace database Version 2 [145, 148]. This database contains 600 face images that can be separated into 200 groups, and each group consists of children, young parents and old parents. Figure 3.6(c) shows example images in the KinFace database, in which three columns (from left to right) represent the images of children, young parents and old parents, respectively. Given two images of faces, our task is to determine whether they are an accurate child-parent pair.

As suggested in [148], we employ the difference vectors between the child and the parent as the features rather than directly compare children with their parents. Specifically, in the experiments for children and old parents, we build 200 true child-old parent pairs and 200 false child-old parent pairs. The experiments for children and young parents are carried out in a similar manner. Then we
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

Table 3.6: Verification rates (%) on UB KinFace database (5-fold cross validation). “C vs. Y” and “C vs. O” denote child-young parent verification and child-old parent verification, respectively.

<table>
<thead>
<tr>
<th>Methods</th>
<th>C vs. Y</th>
<th>C vs. O</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA [116]</td>
<td>57.25±2.59</td>
<td>56.75±2.05</td>
</tr>
<tr>
<td>LDA [6]</td>
<td>47.58±5.36</td>
<td>49.25±7.27</td>
</tr>
<tr>
<td>NPE [118]</td>
<td>55.25±3.01</td>
<td>55.75±3.60</td>
</tr>
<tr>
<td>LSDA [9]</td>
<td>56.75±3.24</td>
<td>57.00±2.88</td>
</tr>
<tr>
<td>RPCA+LDA</td>
<td>56.25±5.08</td>
<td>52.00±4.56</td>
</tr>
<tr>
<td>SVM [146]</td>
<td>51.00±2.05</td>
<td>49.25±1.43</td>
</tr>
<tr>
<td>FDDL [147]</td>
<td>48.00±2.21</td>
<td>51.46±3.51</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>55.00±3.53</td>
<td>52.50±2.76</td>
</tr>
<tr>
<td>DLRD [38]</td>
<td>53.63±3.19</td>
<td>54.27±2.88</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>61.79±2.13</strong></td>
<td><strong>62.15±2.23</strong></td>
</tr>
</tbody>
</table>

In each round, 160 true pairs and 160 false pairs are used for training, and the rest are used for testing. Average verification rates are reported in Table 3.6. Our approach outperforms all the other methods. In this binary classification problem, some traditional supervised methods perform very poorly.

3.3.5 Discussions

The experimental results show that, compared with traditional subspace learning methods, our approach is robust to noise and large variations. The reason is that low-rank property helps us obtain a better estimate of the underlying distribution of samples from the recovered images, and then our approach learns a robust and discriminative subspace. The resulting performance is better than the compared low-rank modeling and dictionary learning methods.

Figure 3.8 illustrates why our approach performs so well by visualizing low-rank representation coefficients of LatLRR and our approach. In particular, we show the coefficients for representing two pairs of samples. One pair, samples A and B, is selected from the same class, while the other pair, samples A and C, from different classes. Figure 3.8(a) and (b) show that, in LatLRR, samples from the same class contribute more in the representation, as the coefficients within the same class are a little larger than those in other classes. In some sense, LatLRR could discover the subspace membership of samples. Compared with LatLRR, Figure 3.8(c) and (d) show that the coefficients of same class are higher than others, which implies our approach can clearly reveal the subspace structure. Since we incorporate supervised regularization in our model, the low-rank representations as well as the resulting subspace learnt by our approach should be more discriminative than that of
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

(a) COIL (b) ALOI (c) FERET (d) AR (e) YaleB

Figure 3.9: Visualization of corrupted images ($X$), recovered images ($AZ$) and noise ($E$) on five object and face databases.

Furthermore, Figure 3.9 visualizes the corrupted images, recovered images and the noisy part on five object and face databases. It shows that, although training images (i.e., $X$) have large pose variations and corruptions, the recovered images $AZ$ are very similar to each other, which helps us learn a robust subspace for classification.

We evaluate the computational cost of different methods when increasing the sample size. Taking COIL database as an example, the training times are shown in Figure 3.10. Since PCA, LDA, NPE and LSDA have similar computational complexity, and FDDL has a similar complexity to DLRD, we only compare against LatLRR, PCA and DLRD. In Figure 3.10, we observe that linear subspace method PCA has the lowest training time. Our approach and LatLRR have similar training
CHAPTER 3. ROBUST SUBSPACE DISCOVERY

time, which is much less than the time cost of DLRD. Moreover, the test time of our approach is even less than that of PCA, due to the fact that our approach can achieve the best recognition rates with only a few features.

3.4 Summary

In this chapter, a novel linear subspace learning approach, supervised regularization based robust subspace (SRRS), is proposed for feature extraction and classification. The proposed approach iteratively learns robust subspaces from a low-rank learning model, and naturally incorporates discriminative information. The convexity of the supervised regularization term has been theoretically proven. Experimental results on six benchmark datasets demonstrate the effectiveness of our approach compared with the state-of-the-art subspace methods and low-rank learning methods. Moreover, when the data contain considerable noise or variations, our approach can improve the classification performance.

In our future work, we will develop a divide-and-conquer version of SRRS approach to make it scalable to larger datasets, and we would also like to design dictionary learning algorithms to further enhance the classification performance.
Chapter 4

Robust Multi-View Subspace Learning

4.1 Background and Motivation

Nowadays information about one object can be continuously collected from multiple views in many domains such as health care and entertainment, due to the increasingly large amount of various sensors. The collected data can be represented as multi-view multivariate time series, which could lead to significant improvement of data mining tasks like classification. For instance, the daily activities of a subject can be captured by video cameras, depth cameras, and on-body sensors. These three sets of heterogeneous and dynamic measurements would provide mutually enriched information, which are helpful for improving the performance of activity recognition. In general, it has been well recognized that multi-view data usually enhance the overall model performance than single-view data, as long as the different views contain diverse information [149]. In this chapter, we focus on the classification of multi-view multivariate time series, which plays a central role in extracting useful knowledge from the multi-view streaming data.

Although the classification of time-series data has been extensively studied during the past decade, they are only designed for single-view data. Traditional methods focus on the univariate time series (u.t.s.) classification [150, 151], by defining distance measures (e.g., dynamic temporal wrapping (DTW) [152], recurrence plot [153], edit distance [154] and elastic distance [155]), or extracting compact and effective features (e.g., time series shapelets [156] and segment based features [157]). Furthermore, as the streaming data might be characterized by multiple measurements simultaneously and represented as multivariate time series (m.t.s.), some recent works try to extract informative patterns from m.t.s., and have achieved promising results [158, 159, 160, 161]. They can be roughly categorized into three groups: (1) distance metric; (2) classifier design; (3) dimensionality
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

reduction. The first group of methods focus on designing distance metrics for m.t.s., by considering the temporal dynamics and the possible misalignment problem in m.t.s. [159]. The classifier design methods usually adapt the effective classifiers from other domains to m.t.s. classification, such as SVM [162], recurrent probabilistic neural network [163], convolutional nonlinear component analysis [160], etc.. The dimensionality reduction methods project high-dimensional m.t.s. to a low-dimensional subspace by satisfying certain criteria. Weng et al. employed two-dimensional singular value decomposition [164] and locality preserving projections [165] for m.t.s. classification. Li et al. designed a common principal component analysis method for m.t.s. classification. Other interesting explorations on m.t.s. classification include feature selection [166], temporal abstraction [167], and tensor factorization [168]. However, these methods cannot directly handle multi-view data, and the benefits of mutual-support multiple views are not taken into account.

On the other hand, multi-view learning has attracted increasing attention in recent years [42, 169], since it sophisticatedly models the consistency and diversity among multiple data views, and significantly boosts the learning performance than single-view methods. The popular multi-view learning algorithms are usually categorized as co-training, multiple kernel learning, and subspace learning [42]. However, existing multi-view algorithms are not customized for time series classification, as they simply ignore the unique properties of time series, such as the temporal smoothness.

To address the above challenges, we propose a novel approach, named Multi-view Discriminative Bilinear Projections (MDBP), for multi-view m.t.s. classification [170]. Figure 4.1 illustrates the framework of our approach. MDBP aims to extract discriminative features from multi-view m.t.s. data, and it models the view consistency and temporal dynamics. First, we assume that a m.t.s. sample and its counterparts observed in other views could share a compact representation in a low-dimensional latent subspace, as they indeed represent the same object. To preserve the original temporal structure in m.t.s., MDBP learns a pair of view-specific bilinear projections, which separately reduce the dimensions of measurements and timestamps. Second, MDBP enforces that samples belong to the same class share the same latent representation in the shared subspace, which reduces the within-class separability. Also, the latent representations of different classes are pushed away from each other, in order to enhance the between-class separability. Third, a Laplacian regularization term is designed to preserve the temporal smoothness of m.t.s. after projection. An efficient optimization algorithm based on gradient descent is designed to solve the problem. We evaluate the classification performance of our approach and baseline methods on two real-world datasets including the UCI Daily and Sports Activity dataset, and the Multimodal Spoken Word dataset. Extensive results on both single-view and multi-view test scenarios demonstrate the superiority of
Figure 4.1: Framework of our MDBP approach. The Multimodal Spoken Word dataset contains three data views, including video, audio, and magnetic sensors. MDBP maps multi-view data onto a shared subspace through a pair of view-specific bilinear projections (i.e., $P_1$ and $Q_1$), and incorporates discriminative information by enhancing the between-class separability. $Y_i$ is the compact representation of $i$-th class in the latent space.

Our approach greatly improves the classification accuracy due to the full exploration of multi-view m.t.s. data.

Our work is closely related to the research topics of subspace learning, multi-variate time series classification, and multi-view learning. Subspace learning methods reduce the dimensionality of data through linear or nonlinear projections [1, 3, 171, 5]. The major differences between our approach and existing subspace learning methods are: (1) our approach deals with the multi-view time series data by modeling view consistency and temporal regularization; (2) our approach incorporates a novel supervised regularization for classification tasks. Multi-view learning aims to extract shared knowledge from multiple data sources [42, 43, 44, 47, 48, 46]. However, existing multi-view learning algorithms do not take the temporal information into account, which are not suitable for m.t.s. classification. Our MDBP approach incorporates a temporal smoothness regularization, and its effectiveness has been validated by extensive experiments.

The main contributions of this chapter are summarized as follows.

- We propose a discriminative bilinear projection approach, MDBP, for multi-view multivariate time-series classification. To the best of our knowledge, this work is the first attempt to apply
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

Table 4.1: Notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{vij}$</td>
<td>The $j$-th m.t.s. sample in $i$-class, $v$-th view</td>
</tr>
<tr>
<td>$P_v$</td>
<td>Bilinear projection for the $v$-th view</td>
</tr>
<tr>
<td>$Q_v$</td>
<td>Bilinear projection for the $v$-th view</td>
</tr>
<tr>
<td>$Y_i$</td>
<td>Representation of $i$-th class in latent space</td>
</tr>
<tr>
<td>$L_p$</td>
<td>Temporal Laplacian matrix</td>
</tr>
<tr>
<td>$V$</td>
<td>Number of views</td>
</tr>
<tr>
<td>$C$</td>
<td>Number of classes</td>
</tr>
<tr>
<td>$N_i$</td>
<td>Number of samples in the $i$-th class</td>
</tr>
<tr>
<td>$d_v$</td>
<td>Dimensionality of m.t.s. sample in $v$-th view</td>
</tr>
<tr>
<td>$m_v$</td>
<td>Length of m.t.s. sample in $v$-th view</td>
</tr>
</tbody>
</table>

multi-view dimensionality reduction for the m.t.s. classification problem.

- We model the view consistency by projecting multi-view m.t.s. onto a shared subspace, and incorporate the discriminative regularization and temporal smoothness regularization.

- We conduct extensive experiments on two real-world datasets, which demonstrate the effectiveness of our approach, compared to the state-of-the-art multi-view learning methods and m.t.s. classification methods.

4.2 Problem Definition

In this section, we first present the definitions of multivariate time series (m.t.s.) and multi-view m.t.s., and then formally define the problems of m.t.s. classification and multi-view m.t.s. classification. Table 4.1 summarizes the notations used throughout this chapter.

Definition 1. Multivariate Time Series (m.t.s.). A multivariate time series (m.t.s.) $X = [x_1, \cdots, x_m] \in \mathbb{R}^{d \times m}$ is an ordered sequence of $d$-dimensional vectors, in which $x_i$ is the observation at the $i$-th timestamp, and $m$ is the length of time series.

Nowadays time-series data are usually collected from multiple views or multiple modalities. We define the multi-view multivariate time series as follows.

Definition 2. Multi-View Multivariate Time Series (m.t.s.). A multi-view m.t.s. $\hat{X} = \{X_{(v)}\}$, $v = 1, \cdots, V$ is a set of time series data collected from multiple views, where $X_{(v)} \in \mathbb{R}^{d_v \times m_v}$ denotes the time series observed in the $v$-th view, $d_v$ is the number of measurements of time series, $m_v$ is the length of time series, and $V$ is the total number of views.
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

For simplicity, we assume that the time series within the same view have been synchronized and preprocessed, and therefore they have the same length $m_v$.

We focus on the classification task in this chapter. The formal definition of multivariate time series classification is as follows.

**Definition 3. Multivariate Time Series (m.t.s.) Classification.** Let $\mathcal{C}$ denote a set of class labels, and $|\mathcal{C}|$ is the total number of classes. The task of m.t.s. classification is to learn a classifier, which is a function $F : \mathbf{X} \rightarrow \mathcal{C}$, where $\mathbf{X}$ is a set of m.t.s.

The traditional m.t.s. classification algorithms are mainly designed for single-view data, and they cannot directly handle the multi-view data. Although some practical tricks might be adopted, such as vectorizing multi-view data to single-view ones, we argue that considerable information might be discarded during this process. In this chapter, we extend the single-view m.t.s. classification problem to the multi-view setting, and present the formal definition as follows.

**Definition 4. Multi-View Multivariate Time Series Classification.** Let $\hat{\mathbf{X}} = \{X_{i,v} | i = 1, \cdots, N, v = 1, \cdots, V\}$ denote a set of multi-view m.t.s., where $N$ is the number of m.t.s. in each view, and $\mathcal{C} = \{C_1, \cdots, C_c\}$ denote a set of class labels shared by $V$ views. The task of multi-view m.t.s. classification is to learn a classifier from $\hat{\mathbf{X}}$, and therefore to infer the class label for test m.t.s. $X_{\text{test}}$ which might be observed in any view.

We notice that the basic m.t.s. classification can be considered as a special case of multi-view m.t.s. classification when there is only one view available. In the multi-view case, the m.t.s. classification problem becomes more challenging. For example, the consistency between multiple views should be modeled.

### 4.3 Multi-view Discriminative Bilinear Projection (MDBP)

In this section, we propose the multi-view discriminative bilinear projections (MDBP) approach for m.t.s. classification. We first introduce our motivation, and then present the model details. Finally, the optimization algorithm with discussions is provided.

#### 4.3.1 Motivation

For multi-view m.t.s. classification, several key problems should be taken into account, including:

1. **Synchronization and Preprocessing.**
2. **Modeling Multiple Views.**
3. **Consistency Between Views.**
4. **Dimensionality Reduction.**
5. **Classification Accuracy.**

The MDBP approach aims to effectively address these challenges by leveraging the discriminative power of bilinear projections in a multi-view setting.
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

(1) How to build the consistency and interactions of m.t.s. from multiple views?

(2) How to extract discriminative features from multi-view m.t.s.?

(3) As the data size has to be increased in the multi-view case, how to improve the computational efficiency of training and test?

We aim to address all of the above challenges by designing a multi-view dimensionality reduction approach. The basic idea is to project multi-view data onto a common subspace, and then perform the classification of m.t.s. using the low-dimensional representations. Our motivations of using dimensionality reduction are three-folds. First, as multi-view data are usually drawn from diverse data spaces, seeking common projections would allow us to bridge the gap between multiple data views. Learning a shared subspace is also considered as a popular strategy in multi-view learning [43, 172]. Second, specifically designed regularization functions, such as discriminative regularization, can be naturally incorporated into the multi-view dimensionality reduction framework. Third, we aim to learn linear projections for each view, and therefore, the training and test would be efficient.

4.3.2 Formulation of MDBP

We assume that a set of m.t.s. $\hat{X}$ observed from $V$ views belong to $C$ different classes, $\hat{X} = \{X_{vij} | v = 1, \cdots, V, i = 1, \cdots, C, j = 1, \cdots, N_i \}$, where $N_i$ is the number of samples in the $i$-th class for each view.

The general formulation of MDBP is

$$\min_{P, Q, Y} \Phi(\hat{X}, P, Q, Y) + \lambda_1 \Theta(Y) + \lambda_2 \Omega(P, X),$$

where $P$ and $Q$ are bilinear projections, and $Y$ is the low-dimensional representation of $\hat{X}$. $\lambda_1$ and $\lambda_2$ are two trade-off parameters that balance the effects of different terms. Eq (4.1) contains three components. The first term $\Phi(\hat{X}, P, Q, Y)$ represents the multi-view bilinear dimensionality reduction, which characterizes the connections between different views. The second term $\Theta(Y)$ carries discriminative regularization, and the last term $\Omega(P, X)$ models the temporal smoothness. We will detail the three components in the following.

Learning Shared Representations Across Views

We propose to learn bilinear projections for reducing the dimensionality of m.t.s.. Let $P_v \in \mathbb{R}^{d_v \times p}$ and $Q_v \in \mathbb{R}^{m_v \times q}$ denote a pair of linear projections for the $v$-th view, and then the
m.t.s. $X_{vij}$ can be transformed by

$$Y_{vij} = P^\top_v X_{vij} Q_v,$$

(4.2)

where $Y_{vij} \in \mathbb{R}^{p \times q}$ is the low-dimensional representation of $X_{vij}$.

The major benefits of employing bilinear projections are two-folds. First, bilinear projections allow us to preserve the original structure of m.t.s., especially the temporal structures, which makes it easier to incorporate temporal smoothness regularizations along the time dimension. Second, compared to other dimensionality reduction methods, bilinear projections have less computational cost for both training and test, which is suitable for dealing with long-duration time series data.

Eq. (4.2) assumes that each view shares a pair of linear projections. However, it doesn’t take view correlation into account. A more reasonable assumption is that, a sample and its counterparts collected from other views could have the same low-dimensional representation in a common subspace. Moreover, as we focus on classification tasks, we further assume that samples from the same class, no matter which views they belong to, would share approximately the same representations in the common subspace. Therefore, we rewrite Eq. (4.2) as: $Y_i \approx P^\top_v X_{vij} Q_v$, which encourages samples of the same class from all the views to be as close as possible in the common subspace.

Then we formulate the multi-view dimensionality reduction term $\Phi(\hat{X}, P, Q, Y)$ as

$$\Phi(\hat{X}, P, Q, Y) = \sum_{i=1}^{C} \sum_{v=1}^{V} \sum_{j=1}^{N_i} \left\| X_{vij} - P_v Y_i Q_v^\top \right\|_F^2,$$

(4.3)

where $\| \cdot \|_F$ is the matrix Frobenius norm. Here we assume that the projections $P_v$ and $Q_v$ are semi-orthogonal matrices, i.e., $P_v^\top P_v = I_p$ and $Q_v^\top Q_v = I_q$, where $I_p \in \mathbb{R}^{p \times p}$ and $I_q \in \mathbb{R}^{q \times q}$ are two identity matrices.

**Incorporating Discriminative Regularization**

For classification tasks, the learned low-dimensional representations via dimensionality reduction should be discriminative. Actually, $\Phi(\hat{X}, P, Q, Y)$ in Eq. (4.3) already makes use of the label information, as it maps the same-class samples onto a stationary point in the low-dimensional common space. It implicitly incorporates discriminative information, however, the separability among classes hasn’t been included, which is a key for classification problems as suggested by the Fisher criterion [3].

Therefore, to explicitly incorporate the discriminative information, we push the low-dimensional representations of different classes, $Y_i$ and $Y_k$ ($i \neq k$), far away from each other. The
discriminative regularization term $\Theta(Y)$ is defined as

$$\Theta(Y) = - \sum_{i=1}^{C} \sum_{k=1, k \neq i}^{C} \|Y_i - Y_k\|_F^2.$$  \hspace{1cm} (4.4)

As we need to maximize the summation of pairwise distances between $Y_i$ and $Y_k$, a negative sign is added in order to use $\Theta(Y)$ in the minimization problem Eq. (4.1).

The discriminative regularization shown in Eq.(4.4) is view-independent, as it is implemented in the shared subspace. This strategy not only simplifies the model complexity, but also closely relates to the final classification task that is usually performed in the low-dimensional subspace.

Modeling Temporal Smoothness

In reality, many types of time series data, such as human activities, slightly change in successive timestamps, such as the smooth transitions of human activities [61]. In other words, time series data own the property of locally smoothness, which brings informative prior knowledge for learning models. By using bilinear projections, our model does not break the temporal structures of input time series $X$, in which the temporal smoothness is usually observed. However, after projecting $X$ to a low-dimensional subspace via $P_v$, the temporal smoothness might be undermined in the projected data $P_vX$.

To address this problem, we aim to design a smoothness regularization term on $P_vX_{vk}$, where $X_{vk}$ is the $k$-th sample in the $v$-th view. In light of the Laplacian regularization [5], we propose a multi-view temporal Laplacian regularization $\Omega(P_v, X_{vk})$ to enforce the smoothness as follows

$$\Omega(P_v, X_{vk}) = \frac{1}{2} \sum_{i,j=1}^{N} W_{ij} \left\| P_v^\top X_{vk(i)} - P_v^\top X_{vk(j)} \right\|_2^2$$

$$= \sum_{i=1}^{N} P_v^\top X_{vk(i)} D_{ii} X_{vk(i)}^\top P_v - \sum_{i,j=1}^{N} P_v^\top X_{vk(i)} W_{ij} X_{vk(j)}^\top P_v$$

$$= \text{tr}(P_v^\top X_{vk} D X_{vk}^\top P_v) - \sum_{i,j=1}^{N} P_v^\top X_{vk(i)} W_{ij} X_{vk(j)}^\top P_v$$

$$= \text{tr}(P_v^\top X_{vk}(D - W) X_{vk}^\top P_v)$$

$$= \text{tr}(P_v^\top X_{vk}(L_P) X_{vk}^\top P_v),$$  \hspace{1cm} (4.5)

where $X_{vk(i)} \in \mathbb{R}^{d \times 1}$ is the $i$-th column in $X_{vk}$, $\text{tr}(\cdot)$ denotes the trace of a matrix, $W$ is a pre-defined weight matrix that carries the smoothness prior, $D$ is a diagonal matrix whose entries are $D_{ii} = \sum_j W_{ij}$, and $L_P(= D - W)$ is the Laplacian matrix.
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

Let \( Z_{vk} \) denote the projected feature of \( X_{vk} \), \( Z_{vk} = P_v^\top X_{vk} \). It is clear that each column in \( X_{vk} \) or \( Z_{vk} \) corresponds to a timestamp. In reality, successive neighbors in \( X_{vk} \) usually slightly change over time, which can be considered as prior information of temporal smoothness. By setting a proper weighting matrix \( W \), we can transfer such temporal smoothness from \( X_{vk} \) to \( Z_{vk} \) using the Laplacian regularization \( \Omega(P_v, X_{vk}) \). Let \( s \) denote the number of successive neighbors, the entry in \( W \) is computed as

\[
W_{ij} = \begin{cases} 
1, & \text{if } |i - j| \leq \frac{s}{2} \\
0, & \text{otherwise}
\end{cases}
\] (4.6)

In this way, the successive columns in \( Z_{vk} \) are encouraged to be similar to each other. Note that we only adopt binary weights in Eq. (4.6). Other sophisticated graph weighting schemes could also be employed to construct \( W \).

Then, the regularization term \( \Omega(P, X) \) used in Eq. (4.1) is defined as a summation of \( \Omega(P_v, X_{vk}) \) over all of the views and samples

\[
\Omega(P, X) = \sum_{v=1}^{V} \sum_{k=1}^{N} \Omega(P_v, X_{vk}).
\] (4.7)

Objective Function

To sum up, the objective function of our MDBP approach is:

\[
\min_{P_v, Q_v, Y_i} f(P_v, Q_v, Y_i) = \sum_{i=1}^{C} \sum_{v=1}^{V} \sum_{j=1}^{N_i} \| X_{vij} - P_vY_iQ_v^\top \|_F^2 \\
- \lambda_1 \sum_{i=1}^{C} \sum_{k=1, k \neq i}^{C} \| Y_i - Y_k \|_F^2 + \lambda_2 \sum_{v=1}^{V} \sum_{k=1}^{N} \text{tr}(P_v^\top X_{vk}(L_P)X_{vk}^\top P_v) \\
\text{s.t. } P_v^\top P_v = I_p, \quad Q_v^\top Q_v = I_q, v = 1, \ldots, V.
\] (4.8)

In Eq. (4.8), orthogonal constraints \( P_v^\top P_v = I_p \) and \( Q_v^\top Q_v = I_q \) are incorporated. Orthogonality in a projection matrix means that any two basis vectors in this projection are orthogonal to each other, which has the advantages of compactness and reducing redundancy.

4.3.3 Optimization Algorithm

We develop an efficient optimization algorithm based on gradient descent to solve the problem in Eq. (4.8).

Although Eq. (4.8) is not jointly convex to all the variables \( P_v, Q_v \) and \( Y_i \), it is convex to each of them when the other variables are fixed. We use gradient descent to alternately update each...
variable. Given $P_v^{(t)}, Q_v^{(t)}, Y_i^{(t)}$ obtained in the $t$-th step, the update rules at the $t + 1$ step are

$$P_v^{(t+1)} = P_v^{(t)} - \gamma \frac{\partial}{\partial P_v} f(P_v, Q_v, Y_i), \quad v = 1, \cdots, V,$$

$$Q_v^{(t+1)} = Q_v^{(t)} - \gamma \frac{\partial}{\partial Q_v} f(P_v, Q_v, Y_i), \quad v = 1, \cdots, V,$$

$$Y_i^{(t+1)} = Y_i^{(t)} - \gamma \frac{\partial}{\partial Y_i} f(P_v, Q_v, Y_i), \quad i = 1, \cdots, C,$$

where $\gamma$ is the learning rate.

The detailed derivatives are shown below

$$\frac{\partial}{\partial P_v} = - \sum_{i=1}^{C} \sum_{v=1}^{V} \sum_{j=1}^{N_i} 2(X_{vij} - P_v Y_i Q_v^\top)Q_v Y_i^\top + \lambda \sum_{k=1}^{N} 2P_v^\top X_{vk}(L_P) X_{vk}^\top.$$

(4.12)

$$\frac{\partial}{\partial Q_v} = - \sum_{i=1}^{C} \sum_{v=1}^{V} \sum_{j=1}^{N_i} 2Y_i^\top P_v^\top (X_{vij} - P_v Y_i Q_v^\top).$$

(4.13)

$$\frac{\partial}{\partial Y_i} = - \sum_{i=1}^{C} \sum_{j=1}^{N_i} 2P_v^\top (X_{vij} - P_v Y_i Q_v^\top) Q_v - \lambda \sum_{k=1}^{C} \sum_{j=1}^{N_i} 2(Y_i - Y_k).$$

(4.14)

Note that the orthogonal constraints shown in Eq. (4.8) are implemented by a post-processing step during the update. The complete optimization algorithm is summarized in Algorithm 1. We will show the convergence property of our algorithm in Section 4.4.3.

After obtaining the subspaces $P_v$ and $Q_v$, the nearest neighbor classifier can be employed to classify a test m.t.s. $T_v$. The complete procedures of MDBP are provided in Algorithm 2.

**Time Complexity Analysis**

The computational cost of Algorithm 1 mainly depends on the Step 6, Step 8, and Step 12, which cost $O(N dpq + dqm)$, $O(N dpq + dqm)$, and $O(N pdm + pmq)$, respectively. Indeed, our algorithm reduces the dimensionality of time series, which means $p \ll d$ and $q \ll m$. Thus, the overall time complexity of the three steps is simplified to $O(N (dm + m^2))$.

In addition, our algorithm converges well after several iterations, and there are usually a few views in reality. It indicates that our approach is approximately linear to the sample size $N$ when $N \gg \max(d, m)$, and therefore, our approach can be easily deployed for large-scale applications.

### 4.3.4 Comparison with Existing Methods

The first term in Eq. (4.1), $\sum_{i=1}^{C} \sum_{v=1}^{V} \sum_{j=1}^{N_i} \|X_{vij} - P_v Y_i Q_v^\top\|_F^2$, looks similar to the formulation of matrix tri-factorization [173], which also factorizes a data matrix into three unknown components. However, our approach is motivated from the multi-view learning scenario, and the
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

Algorithm 1 Solving Problem in Eq. (4.8)

**Input:** Multi-view m.t.s. sample set $\tilde{X}$, parameters $\lambda_1, \lambda_2, s, \gamma, \text{maxIter}$.

**Output:** Bilinear projections $P_v, Q_v$

class-specific shared representation $Y_i$.

1: Compute the Laplacian matrix $L_p$ according to Eq. (4.5) and Eq. (4.6);
2: Initialize $P_v, Q_v$ and $Y_i$ with random matrices;
3: **for** loop $t$ from 1 to $\text{maxIter}$ do
4:     **for** view $v$ from 1 to $V$ do
5:         Update projection $P_v$ using Eq. (4.9);
6:         Orthogonalize $P_v$;
7:         Update projection $Q_v$ using Eq. (4.10);
8:         Orthogonalize $Q_v$;
9:     **end for**
10:    **for** class $i$ from 1 to $C$ do
11:         Update latent presentation $Y_i$ using Eq. (4.11);
12:    **end for**
13:    **if** the objective converges **then**
14:        Return $P_v, Q_v$ and $Y_i$.
15:    **end if**
16: **end for**

factorized components carry consistency constraints across views or across classes. For instance, the view-specific projection $P_v$ is shared by every sample in the $v$-th view.

Although some existing multi-view learning algorithms also project multi-view data to a common subspace [43, 172, 46], our approach differs from them in that: (1) we employ the bilinear projections to map high-dimensional m.t.s. to a shared low-dimensional subspace; (2) we design a novel discriminative regularization term for multi-view dimensionality reduction. (3) we focus on the time series data classification, and design a Laplacian regularization term to enforce the temporal smoothness.
Algorithm 2 MDBP Approach

Input: Multi-view m.t.s. training sample set $\hat{X}$, single-view m.t.s. test sample $T_v$.

Output: Predicted class label $c_t$ for $T_v$.

1: Normalize each time series sample;
2: Calculate the projections $P_v$ and $Q_v$ using Algorithm 1;
3: Project $X_{vi}$, $i = 1, \cdots, N$, to the shared subspace by $Z_{vi} = P_v^T X_{vi} Q_v$;
4: Project $T_v$ to the shared subspace by $\hat{Z}_v = P_v^T T_v Q_v$;
5: Predict the class label of $T_v$ using NN classifier, by comparing $\hat{Z}_v$ with $Z_{vi}$.

4.4 Experiments

In this section, we conduct extensive experiments to evaluate the classification performance of our approach and baseline methods on two datasets, and perform quantitative analysis on parameter sensitivity.

4.4.1 UCI Daily and Sports Activity Dataset

The UCI Daily and Sports Activity Dataset [174, 175] contains motion sensor data of 19 daily and sports activities, such as sitting, standing, walking, running, jumping, etc. Each activity is performed by 8 subjects (4 female and 4 male, between the ages 20 and 30) for 5 minutes. In particular, the subjects are asked to perform these activities in their own styles without any restrictions. As a result, the time series samples for each activity have considerable inter-subject variations in terms of speed and amplitude, which makes it difficult for accurate classification. During the data collection, nine sensors are put on each of the following five units: torso, right arm, left arm, right leg, and left leg. Thus, there are 45 sensors in total, and each sensor is calibrated to acquire data at 25 Hz sampling frequency. The 5-minute time series collected from each subject is divided into 5-second segments. For each activity, the total number of segments is 480, and each segment is considered as a m.t.s. sample of size $45 \times 125$, corresponding to 45 sensors and 125 timestamps.

Two-View Setting

We design a two-view experimental setting on the UCI Daily and Sports Activity dataset. Specifically, the first 27 sensors on torso, right arm and left arm are treated as View-1, while the rest 18 sensors on right leg and left leg as View-2. The activities are observed from two distinct views.
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

(i.e., two groups of sensors) simultaneously. Also, the m.t.s. samples in two views have the same number of timestamps.

Baselines

Our MDBP approach is a multi-view dimensionality reduction method for time series classification. We mainly compare it with single-view and multi-view dimensionality reduction methods. The single-view methods include principal component analysis (PCA) [1], linear discriminant analysis (LDA) [3], locality preserving projections (LPP) [165], and two-dimensional LDA (2DLDA) [176]. The multi-view methods include canonical correlation analysis (CCA) [50] and multi-view discriminant analysis (MvDA) [46]. In addition, we also compare our approach with a popular classification method, support vector machine (SVM) [177], and the state-of-the-art time series classification method, one-nearest-neighbor dynamic time warping (1NN-DTW) [152]. For all the baselines except 2DLDA, we have to vectorize each m.t.s. sample into a single vector. Our approach and 2DLDA learn linear projections without vectorizing m.t.s..

Classification Scheme

Given a test time series sample $T_v \in \mathbb{R}^{d \times m}$ that is only observed in the $v$-th view, our approach maps it to a low-dimensional subspace using the learned projections $P_v$ and $Q_v$. Then we employ the nearest neighbor (NN) classifier to assign a class label to $T_v$.

In addition, if a test time series and its counterparts are available in multiple views, we could map them to multiple subspaces using the corresponding bilinear projections $P_v$ and $Q_v$, and then perform NN classification by adopting a feature fusion strategy. We will evaluate the multi-view test cases in Section 4.4.3.

Results

There are 480 samples for each activity per view. We randomly choose $N_{tr}$ samples from each activity (per view) to construct the training set, and the remaining samples are used to construct the test set. In particular, $N_{tr} \in \{10, 20, 30, 40, 50\}$. For singe-view baselines, we separately train two models on the training sets of two views, and report the classification accuracy on each view. For multi-view methods, we train the model by jointly using samples from two views, and also report the accuracy on each view. The parameters in our approach and baselines are tuned using 5-fold cross validation on the training set. The learning rate $\gamma$ in our approach is empirically set to 0.01. We will analyze the parameter sensitivity of our approach in Section 4.4.3.

We randomly choose $N_{tr}$ training samples from each activity 10 times, and report the
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

Table 4.2: Classification Accuracy (%) on UCI Daily Activity Dataset. \(N_{tr}\) is the number of training samples randomly chosen from each activity. V1 and V2 denote the View-1 and View-2, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>(N_{tr} = 10)</th>
<th>(N_{tr} = 20)</th>
<th>(N_{tr} = 30)</th>
<th>(N_{tr} = 40)</th>
<th>(N_{tr} = 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V1</td>
<td>V2</td>
<td>V1</td>
<td>V2</td>
<td>V1</td>
</tr>
<tr>
<td>PCA [1]</td>
<td>27.63</td>
<td>21.73</td>
<td>31.17</td>
<td>23.34</td>
<td>32.01</td>
</tr>
<tr>
<td>LDA [3]</td>
<td>31.35</td>
<td>14.29</td>
<td>38.51</td>
<td>13.33</td>
<td>42.27</td>
</tr>
<tr>
<td>SVM [177]</td>
<td>22.32</td>
<td>20.80</td>
<td>21.45</td>
<td>18.05</td>
<td>21.47</td>
</tr>
<tr>
<td>LPP [165]</td>
<td>27.60</td>
<td>21.18</td>
<td>39.96</td>
<td>30.39</td>
<td>48.79</td>
</tr>
<tr>
<td>2DLDA [176]</td>
<td>53.37</td>
<td>55.24</td>
<td>67.59</td>
<td>64.70</td>
<td>73.15</td>
</tr>
<tr>
<td>DTW [152]</td>
<td>41.05</td>
<td>38.53</td>
<td>43.67</td>
<td>40.33</td>
<td>48.92</td>
</tr>
<tr>
<td>CCA [50]</td>
<td>28.36</td>
<td>18.05</td>
<td>43.10</td>
<td>20.02</td>
<td>51.61</td>
</tr>
<tr>
<td>MvDA [46]</td>
<td>56.43</td>
<td>57.98</td>
<td>75.03</td>
<td>74.24</td>
<td>81.20</td>
</tr>
<tr>
<td>MDBP (Ours)</td>
<td><strong>70.29</strong></td>
<td><strong>67.93</strong></td>
<td><strong>82.58</strong></td>
<td><strong>77.31</strong></td>
<td><strong>87.55</strong></td>
</tr>
</tbody>
</table>

average classification accuracy of our approach and baselines in Table 4.2. Our observations are:

- For smaller training sets (e.g., \(N_{tr} = 10\)), the supervised dimensionality reduction methods like LDA usually achieve higher accuracies than unsupervised methods such as PCA and LPP. The reason is that unsupervised methods cannot accurately estimate the data distribution without sufficient sampling, while supervised information used in LDA play a critical role in this scenario. When the training set grows, PCA achieves comparable results than LDA, and LPP outperforms LDA significantly.

- By preserving the original temporal structure of m.t.s. data, 2DLDA obtains the best results among all the single-view methods, but it cannot make use of the complementary information from multiple views.

- The multi-view methods usually perform better than single-view methods. For instance, the unsupervised multi-view method CCA always obtains higher accuracies than PCA and LPP in the case of View-1; the supervised multi-view method MvDA performs best among all the baseline methods, which demonstrates the effectiveness of multi-view learning and supervised regularization.

- Our approach achieves the highest classification accuracy in every case. Compared to MvDA, the accuracy is improved by at least 6% on average. It demonstrates the superiority of incorporating discriminative information and temporal smoothness to multi-view dimensionality reduction.

Choosing a proper dimension plays a key role in various dimensionality reduction methods, which is still an open problem to date. Figure 4.2 shows the classification performance of our
approach and baselines with different dimensions, when $N_{tr}$ is set to 20, on the View-1 of UCI Daily and Sports Activity dataset. It shows that PCA achieves quite stable results when the dimension is higher than 60. LPP achieves its best performance when the dimension is around 50, and CCA favors a higher dimension. Although LDA increases the accuracy significantly with more dimensions, it is limited by the maximum number of dimension which is less than the number of classes. MvDA requires about 200 dimensions to achieve its best performance. Our approach obtains good performance with only 60 dimensions, and it consistently outperforms other baselines in each case.

4.4.2 Multimodal Spoken Word Dataset

The Multimodal Spoken Word dataset is collected to study the speaker-dependent speech recognition problem, which helps us understand the speech translation for assistive communication. One subject is asked to speak 73 words, and each word is repeated for eight times. The speech of every word is recorded by three types of signals, including audio, video, and magnetic sensors. For audio signals, we extract 20 different features from them, such as Linear Predictive Codings (LPC) [178] and Mel-Frequency Cepstral Coefficients (MFCC) [179]. The videos capture the face of speaker during speech. We crop the mouth regions in the video, and extract the Local Binary Pattern (LBP) [180] features from each frame. Twenty-four magnetic sensors are placed on the tongue of the
subject, which track the positions and movement trajectories of the tongue during speech. Clearly, all of the three modalities can be represented as multivariate time series.

### Three-View Setting

A three-view experimental setting is designed on the Multimodal Spoken Word dataset. The m.t.s. of sensors, video, and audio are separately denoted as View-1, View-2 and View-3. We compare our approach with the baselines described in Section 4.4.1. The m.t.s. within each view are preprocessed to have the same length. In addition, MvDA requires that samples in different views should have the same dimension, while our approach does not have such a constraint. For MvDA, we have to perform preprocessing to make sure that samples in three views share the same dimension.

### Results

We randomly choose $N_{tr} \in \{3, 4, 5\}$ samples from each word (per view) to construct the training set, and the remaining samples are used for the test. This process is repeated for 10 times. Table 4.3 shows the average classification accuracy of our approach and baselines in different settings. We observe that traditional subspace learning methods, such as PCA, LDA and LPP, obtain very poor performance on this dataset, due to the small-sample-size problem. Moreover, the classification task on this dataset is more challenging than that on the UCI Daily and Sports Activity dataset, as there are more classes. 2DLDA keeps the temporal structure of raw m.t.s., and therefore it outperforms other single-view methods. MvDA obtains poor performance on View-2 and View-3, due to the following reasons: (1) MvDA constructs joint scatter matrices across different views, which works well on multi-view data with similar types of features in each view, such as the UCI dataset used in Section 4.4.1. However, the Multimodal Spoken Word dataset contains three different types of

Table 4.3: Classification accuracy (%) on Multimodal Spoken Word dataset. $N_{tr}$ is the number of training samples randomly chosen from each word. V1, V2 and V3 denote the View-1, View-2 and View-3, respectively. SV and MV denote the single-view and multi-view methods, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>$N_{tr} = 3$</th>
<th></th>
<th></th>
<th>$N_{tr} = 4$</th>
<th></th>
<th></th>
<th>$N_{tr} = 5$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V1</td>
<td>V2</td>
<td>V3</td>
<td>V1</td>
<td>V2</td>
<td>V3</td>
<td>V1</td>
<td>V2</td>
<td>V3</td>
</tr>
<tr>
<td>PCA [1]</td>
<td>17.73</td>
<td>17.10</td>
<td>12.47</td>
<td>18.49</td>
<td>17.67</td>
<td>13.42</td>
<td>19.36</td>
<td>17.58</td>
<td>13.70</td>
</tr>
<tr>
<td>2DLDA [176]</td>
<td>50.08</td>
<td>64.66</td>
<td>21.15</td>
<td>55.27</td>
<td>69.04</td>
<td>37.36</td>
<td>62.83</td>
<td>71.69</td>
<td>50.55</td>
</tr>
<tr>
<td>DTW [152]</td>
<td>53.71</td>
<td>65.29</td>
<td>25.45</td>
<td>59.59</td>
<td>58.90</td>
<td>38.47</td>
<td>65.20</td>
<td>72.05</td>
<td>52.33</td>
</tr>
<tr>
<td>MvDA [46]</td>
<td>49.73</td>
<td>39.97</td>
<td>18.75</td>
<td>49.93</td>
<td>38.15</td>
<td>23.20</td>
<td>44.02</td>
<td>32.33</td>
<td>21.25</td>
</tr>
<tr>
<td>MDBP (Ours)</td>
<td>66.44</td>
<td>69.01</td>
<td>39.51</td>
<td>70.24</td>
<td>76.10</td>
<td>41.08</td>
<td>73.01</td>
<td>78.36</td>
<td>61.14</td>
</tr>
</tbody>
</table>
s, which can hardly be characterized by a joint scatter matrix. (2) MvDA requires that samples in different views should have the same dimension, which results in certain information loss. (3) MvDA breaks the temporal structure by vectorizing the m.t.s. samples. Table 4.3 shows that our approach achieves consistently better results than baselines on all the three views.

Figure 4.3 shows the accuracy of our approach and baselines with different dimensions when \( N_{tr} \) is set to 3. Our approach obtains higher accuracy than other baselines in most cases.

4.4.3 Discussions

Parameter Sensitivity and Convergence

There are three major parameters in our approach, including \( \lambda_1, \lambda_2 \) and \( s \). The first two balance the effects of discriminative regularization and temporal smoothness regularization, and parameter \( s \) denotes the number of sequential neighbors used to construct the Laplacian matrix. Figure 4.4 shows the sensitivity of \( \lambda_1 \) and \( \lambda_2 \) on the UCI Daily and Sports Activity dataset. We have the following observations: (1) By setting either \( \lambda_1 \) or \( \lambda_2 \) to 0 (i.e., removing the regularization terms in Eq. (4.8)), the accuracy of our approach drops significantly. It validates the effectiveness of incorporating discriminative and temporal information into our approach. (2) Our approach obtains
Figure 4.4: Parameter sensitivity of $\lambda_1$ and $\lambda_2$ in our approach on UCI Daily and Sports Activity dataset (View-1). The indexes from 1 to 13 on x/y axis correspond to parameters \{0, $10^{-4}$, 5 $\times$ 10$^{-4}$, 10$^{-3}$, 5 $\times$ 10$^{-3}$, 10$^{-2}$, 5 $\times$ 10$^{-2}$, 0.1, 0.5, 1, 5, 10, 20\}.

relatively stable performance with the settings $\lambda_1 \in [5 \times 10^{-4}, 1]$ and $\lambda_2 \in [1, 20]$. Figure 4.5(a) shows the sensitivity of parameter $s$. It shows that our approach is not very sensitive to the setting of $s$, and $s = 2$ usually leads to a better performance.

Figure 4.5(b) shows the convergence curve of our approach on the UCI Daily and Sports Activity dataset. Our approach quickly converges with only 25 iterations, which makes it efficient for large-scale applications.

**Experiments with Data Fusion and Feature Fusion**

In the above experiments, we assume that the test m.t.s. is only available in one view, as shown in Table 4.2 and Table 4.3. In practice, however, test m.t.s. might be available in multiple views. For single-view methods, strategies like data fusion and feature fusion can be applied to generate a final prediction of class label. Multi-view methods can adopt the feature fusion strategy. In data fusion, a m.t.s. observed from multiple views are first vectorized, and then concatenated to a long vector. In feature fusion, the compact features are extracted from each view first, and then those feature vectors can be combined.

Table 4.4 shows the accuracy of our approach and baselines using one or two available fusion strategies on the UCI Daily and Sports Activity dataset. Comparing Table 4.4 and Table 4.2, we observe that the accuracies of PCA, SVM, 2DLDA, and MvDA can be improved by fusing data.
4.5 Summary

In this chapter, we propose a multi-view bilinear projection approach named MDBP for classifying m.t.s. that are collected from multiple views. MDBP projects multi-view data to a shared
CHAPTER 4. ROBUST MULTI-VIEW SUBSPACE LEARNING

subspace through view-specific bilinear projections that preserve the temporal structure of m.t.s., and learns discriminative features by incorporating a novel supervised regularization. The temporal smoothness is also modeled in MDBP, with the help of Laplacian regularization. An efficient optimization algorithm based on gradient descent is designed to solve the problem. We conduct extensive experiments on a daily activity benchmark dataset and a recently collected multimodal spoken word dataset. Experimental results show that our approach obtains remarkable improvements over the state-of-the-art multi-view learning and multivariate time-series classification methods. The parameter sensitivity, convergence property and multi-view fusion are also evaluated and discussed. In our future work, we will develop an online version of MDBP to deal with multi-view m.t.s. in a real-time fashion.
Chapter 5

Robust Dictionary Learning for Human Motion Segmentation

5.1 Background and Motivation

Subspace clustering has attracted an increasing attention in recent years, due to its impressive performance in many real-world applications, such as motion segmentation [13], face clustering [14] and digit clustering [15]. The representative subspace clustering methods include sparse subspace clustering (SSC) [14], low-rank representation (LRR) [13], least-square regression (LSR) [16], etc. The key idea in subspace clustering is to learn effective representation codings that are used to construct an affinity matrix. Many algorithms have been proposed to enhance the performance of subspace clustering, by enforcing different constraints on the coefficients [181], or developing scalable implementations [30, 20, 21, 182].

However, with the notable exception of [183], there are few subspace clustering methods focusing on the data with specific properties, such as the time-series data. Generally, existing methods assume that the data points are independently drawn from multiple subspaces. They either model the data points independently [14] or implicitly consider the global structural information in data [13], but neglect the successive relationships that possibly reside in data. In reality, time-series data like videos can be found everywhere [158, 184, 185, 186]. Labelling or manually processing a large amount of videos is expensive and time-consuming. Therefore, it is necessary to devise unsupervised visual learning algorithms to handle the time-series data.

In this chapter, we propose a temporal subspace clustering (TSC) method based on dictio-
CHAPTER 5. ROBUST DICTIONARY LEARNING FOR HUMAN MOTION SEGMENTATION

Figure 5.1: Framework of the proposed approach. We first extract frame-level features from video sequence, and then learn a non-negative dictionary and the representation codings, with the help of temporal regularization. Multiple temporal segments are finally generated via clustering on the codings.

The most relevant work to ours is the ordered subspace clustering (OSC) [183]. It aims at finding the sparse representations for data, and also introduces a penalty term to take care of the sequential data. OSC also presents an “intrinsic segmentation” strategy to automatically find the segmentation boundaries. However, OSC and our approach have significant technique differences.
First, OSC explicitly adds the temporal regularization via $ZR$, where $R$ is a specific constant matrix. But our approach offers a more flexible way to encode temporal information through Laplacian regularization. Secondly, OSC utilizes the sample set itself as the bases for sparse representation, while our approach learns a non-negative dictionary that consists of expressive bases for temporal subspace clustering.

Another relevant topic is temporal clustering, which segments time series data into a set of non-overlapping groups. It can be applied to learning taxonomies of facial behavior, speaker diarization, discovering motion primitives and clustering human actions in videos. By far only a few temporal clustering methods have been developed, such as the extensions of dynamic Bayesian networks (DBNs) [187], $k$-means [188], spectral clustering [189], and maximum-margin temporal clustering [190]. Basically, these temporal clustering methods focus on the post-processing after graph construction, while the above subspace clustering methods focus on learning codings for graph construction. Our TSC approach mainly belongs to the subspace clustering category. In another word, our approach can be easily concatenated to these post-processing methods to further enhance the performance.

The proposed temporal clustering algorithm will play an important role in unsupervised motion analysis [191, 192, 189]. Recently, some methods based on metric learning [193, 194], regression analysis [195] and spatio-temporal kernel [196] have achieved impressive performance on the motion analysis applications. In this chapter, we will evaluate the clustering performance of our approach and baselines on the real-world motion datasets.

In summary, the contributions of this chapter include:

- We design a novel temporal Laplacian regularization function to model the sequential information in time series data. To the best of our knowledge, this chapter presents the first temporal Laplacian regularization for subspace clustering.

- We develop a non-negative dictionary learning method to learn expressive codings for temporal clustering. The non-negative bases in dictionary are especially useful for the human motion data (e.g., action videos) that are usually non-negative values.

- We present an efficient optimization algorithm to jointly learn the non-negative dictionary and expressive codings, which are used for constructing a robust affinity graph.
CHAPTER 5. ROBUST DICTIONARY LEARNING FOR HUMAN MOTION SEGMENTATION

5.2 Temporal Subspace Clustering via Robust Dictionary Learning

5.2.1 Problem Formulation

The conventional \textit{subspace clustering} (or subspace segmentation) problem is defined as follows.

Let \( \bar{X} \) denote a set of data vectors \( \bar{X} = [\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n] \) (each column is a sample) in a \( D \)-dimensional Euclidean space. These data vectors are assumed to be drawn from a union of \( k \) subspaces \( \{S_i\}_{i=1}^k \) of unknown dimensions. \textit{Subspace clustering} [13] aims to cluster all data vectors into their respective subspaces.

Clearly, traditional subspace clustering problem neglects the temporal information in data, which makes it unsuitable for the time series data. Considering the temporal relationship contained in data, we define the \textit{temporal subspace clustering} problem as follows.

Consider a sequence of time-series data \( X = [x_1, x_2, \cdots, x_n] \) (the \( i \)-column is sampled at time \( t_i \)) drawn from a union of \( k \) subspaces of unknown dimensions, \textit{temporal subspace clustering} groups the \( n \) samples into \( m (m \geq k) \) sequential segments, and clusters the \( m \) segments into their respective subspaces.

Given a dictionary (i.e., bases) \( D \in \mathbb{R}^{d \times r} \) and a coding matrix \( Z \in \mathbb{R}^{r \times n} \), the time series data \( X \in \mathbb{R}^{d \times n} \) can be approximately represented as:

\[
X \approx DZ,
\]

where \( d \) is the dimension of samples, \( r \) is the number of bases in dictionary, and \( n \) is the total number of samples (a.k.a., the number of time stamps).

We adopt the least-square regression based formulation for temporal subspace clustering. The objective function is:

\[
\min_{Z,D} \|X - DZ\|_F^2 + \lambda_1 \|Z\|_F^2,
\]

where \( \|Z\|_F \) denotes the Frobenius norm of \( Z \), i.e., \( \|Z\|_F^2 = \sum_{i=1}^r \sum_{j=1}^n Z_{ij}^2 \), and \( \lambda_1 \) is a trade-off parameter.

The first term \( \|X - DZ\|_F^2 \) captures the reconstruction error, and the second term \( \|Z\|_F^2 \) is used to model the global subspace structure in \( X \). Moreover, it has shown that the Frobenius norm is a good choice to enforce the block diagonal structure in \( Z \), which is the key to recovering subspace structures [16].

Inspired by the commonly-used manifold regularization technique [197], we design a temporal Laplacian regularization function \( f_t(Z) \) to incorporate the temporal information in time
series $X$. The $i$-th column in coding matrix $Z$, $z_i$, can be viewed as a new representation for $x_i$. Our motivation is that, the sequential neighbors of $z_i$ (e.g., $z_{i-1}$, $z_{i+1}$) could be close to $z_i$ in the coding space.

**Definition 1. (Temporal Laplacian Regularization)** Given a coding matrix $Z$, the temporal Laplacian regularization function is defined as:

$$f(Z) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \|z_i - z_j\|^2 = \text{tr}(L_T Z^T Z),$$

(5.3)

where $L_T$ is a temporal Laplacian matrix, $L_T = \tilde{D} - W$, $\tilde{D}_{ii} = \sum_{j=1}^{n} w_{ij}$, $W$ is the weight matrix that captures the sequential relationships in $X$. Let $s$ denote the number of sequential neighbors for each sample, the element in $W$ is calculated as

$$w_{ij} = \begin{cases} 1, & \text{if } |i - j| \leq \frac{s}{2}, \\ 0, & \text{otherwise}. \end{cases}$$

(5.4)

Different from existing Laplacian regularization that considers the spatial closeness of all data points, our temporal regularization function $f(Z)$ mainly focuses on the temporal closeness in time series data.

**Example.** To better illustrate why $f(Z)$ is able to encode the temporal information, we show the structure of $W$ in a simple case. If we have $n = 5$ and $s = 2$, the weight matrix $W$ and the temporal Laplacian matrix $L_T$ are:

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad L_T = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Note that we only use the binary weights for $W$ to show the idea. Other sophisticated graph weighting algorithms can also be applied here to attain better performance.

Then, the objective function in (5.2) can be rewritten as:

$$\min_{Z, D} \|X - DZ\|_F^2 + \lambda_1 \|Z\|_F^2 + \lambda_2 f(Z),$$

(5.5)

where $\lambda_2$ is a trade-off parameter to balance different terms.

We can observe that $L_T$ is a special case of Laplacian matrix by enforcing the temporal consistency, which is more suitable for time series data. In this manner, sample $x_i$ and its sequential
neighbors \( \{x_{i-s/2}, \cdots, x_{i+s/2}\} \) are encouraged to have the similar codings \( \{z_{i-s/2}, \cdots, z_{i+s/2}\} \).

We will show that \( f(Z) \) helps us obtain continuous segments from time series data. Therefore, our model is more robust to noise and abnormal events in the temporal space.

Another key factor in subspace clustering is the choice of dictionary. Dictionary learning has attracted a lot of attention [134], but existing subspace clustering methods usually follow the data self-representation strategy, i.e., the data set \( X \) serves as the dictionary. However, when the sampling is insufficient or the data set \( X \) is heavily corrupted, employing \( X \) as dictionary may hinder the clustering performance. Thus, learning an expressive dictionary is necessary.

To address this problem, we introduce the dictionary learning procedure into problem (5.5). Moreover, as time series data in real-world applications (e.g., action videos, human motions) are usually non-negative values, it is reasonable to learn a non-negative dictionary for temporal subspace clustering, i.e., \( D \geq 0 \). Naturally, the coding matrix \( Z \) should also be non-negative, i.e., \( Z \geq 0 \).

After adding the dictionary learning component and two non-negative constraints, we have the temporal subspace clustering (TSC) model as follows:

\[
\begin{align*}
\min_{Z,D} & \quad \|X - DZ\|_F^2 + \lambda_1 \|Z\|_F^2 + \lambda_2 f(Z), \\
\text{s.t.} & \quad Z \geq 0, \quad D \geq 0, \quad \|d_i\|_2^2 \leq 1, \quad i = 1, \cdots, r.
\end{align*}
\] (5.6)

The non-negative constraints \( Z \geq 0 \) and \( D \geq 0 \) ensure that the learned bases and corresponding bases should be non-negative values, and the constraint \( \|d_i\|_2^2 \leq 1 \) controls the model complexity.

### 5.2.2 Optimization

To solve the objective function (5.6), we devise an optimization algorithm based on the alternating direction method of multipliers (ADMM). To facilitate the optimization, we consider an equivalent form of (5.6):

\[
\begin{align*}
\min_{Z,D,U,V} & \quad \|X - UV\|_F^2 + \lambda_1 \|V\|_F^2 + \lambda_2 f(V), \\
\text{s.t.} & \quad U = D, \quad V = Z, \quad Z \geq 0, \quad D \geq 0, \\
& \quad \|d_i\|_2^2 \leq 1, \quad i = 1, \cdots, r.
\end{align*}
\] (5.7)

where \( U \) and \( V \) are auxiliary variables.
CHAPTER 5. ROBUST DICTIONARY LEARNING FOR HUMAN MOTION SEGMENTATION

The augmented Lagrangian of (5.7) is:

\[
L = \frac{1}{2} \|X - UV\|_F^2 + \lambda_1 \|V\|_F^2 + \lambda_2 \text{tr}(V L_T V^\top) \\
+ \langle \Lambda, U - D \rangle + \langle \Pi, V - Z \rangle + \frac{\alpha}{2} \|U - D\|_F^2 \\
+ \|V - Z\|_F^2
\]

(5.8)

s.t. \ Z \geq 0, \ D \geq 0, \ |d_i|_2^2 \leq 1, \ i = 1, \ldots, r,

where \ \Lambda \text{ and } \Pi \text{ are Lagrangian multipliers, and } \alpha \text{ is a penalty parameter.}

The ADMM algorithm for (5.7) is derived by alternatively minimizing \ \mathcal{L} \ \text{with respect to } V, U, Z \text{ and } D.

**Update V when fixing others.** The problem (5.8) becomes:

\[
\min_V \frac{1}{2} \|X - UV\|_F^2 + \lambda_1 \|V\|_F^2 + \lambda_2 \text{tr}(V L_T V^\top) \\
+ \langle \Pi, V - Z \rangle + \frac{\alpha}{2} \|V - Z\|_F^2
\]

(5.9)

By setting the derivative of (5.9) with respect to \ V \ \text{to zero, we have the following equation:}

\[
(U^\top U + (\lambda_1 + \alpha)I)V + \lambda_2 V L_T = U^\top X - \Pi + \alpha Z.
\]

(5.10)

Eq. (5.10) is a standard Sylvester equation, which can be effectively solved using existing tools such as the Bartels-Stewart algorithm [141]. Alternatively, we can vectorize the linear matrix equation (5.10) into:

\[
[I \otimes (U^\top U + (\lambda_1 + \alpha)I) + \lambda_2 L_T \otimes I] \text{vec}(V) \\
= \text{vec}(U^\top X - \Pi + \alpha Z)
\]

(5.11)

where \ \otimes \text{ is the tensor product.}

**Update U when fixing others.** By ignoring the variables that are irrelevant to \ U \text{, we have:}

\[
\min_U \frac{1}{2} \|X - UV\|_F^2 + \langle \Lambda, U - D \rangle + \frac{\alpha}{2} \|U - D\|_F^2
\]

(5.12)

Setting the derivative of (5.12) with respect to \ U \text{ to zero, we have the solution:}

\[
U = (XV^\top - \Lambda + \alpha D)(VV^\top + \alpha I)^{-1}.
\]

(5.13)

**Update Z and D when fixing others.** The update rules are:

\[
Z = F_+(V + \frac{\Pi}{\alpha}),
\]

(5.14)

\[
D = F_+(U + \frac{\Lambda}{\alpha}),
\]

(5.15)
where \((F_\times(A))_{ij} = \max\{A_{ij}, 0\}\), which meets the non-negative requirements for \(D\) and \(Z\). We also normalize each column vector in \(D\) to unit length.

The above process is repeated until convergence. For the non-convex problems or convex problems with multiple blocks, there is no theoretical guarantee for the global convergence of ADMM. However, we can show the convergence property of ADMM under mild conditions, following the analysis in [198].

**Theorem 1.** Let \(\{(V_k, U_k, Z_k, D_k, \Pi_k, \Lambda_k)\}\) be a sequence generated by Algorithm 1 in the \(k\)-th iteration. If the sequences of multipliers \(\{(\Pi_k, \Lambda_k)\}\) is bounded and satisfies

\[
\sum_{k=0}^{\infty} (\|\Pi_{k+1} - \Pi_k\|_F^2 + \|\Lambda_{k+1} - \Lambda_k\|_F^2) < \infty.
\]

Then any accumulation point of the generated sequence \(\{(V_k, U_k, Z_k, D_k, \Pi_k, \Lambda_k)\}\) satisfies the KKT condition of problem (5.7).

The proof will be provided in the supplementary document due to the space limit.

### 5.2.3 Clustering

The coding matrix \(Z\) can be used to construct an affinity graph \(G\) for subspace clustering. In SSC, LRR and LSR, the definition of \(G\) is \(G = \frac{|Z| + |Z^T|}{2}\). However, this graph does not well exploit the intrinsic relationships of within-cluster samples. For time series data, the within-cluster samples (i.e., sequential neighbors) are always highly correlated to each other [93, 35]. Therefore, we can take advantage of this property and devise another similarity measurement to construct \(G\).

\[
G(i, j) = \frac{z_i^T z_j}{\|z_i\|_2 \|z_j\|_2}.
\]

In the experiments, we evaluate both similarity measurements for every baseline, and report the better results. Finally, an effective clustering algorithm, Normalized Cuts [66], is utilized to produce the temporal clustering results. The complete temporal subspace clustering approach is summarized in Algorithm 1.

### 5.2.4 Discussion

Note that when setting \(\lambda_1 = \lambda_2 = 0\) and removing the constraints \(\|d_i\|_2^2 \leq 1\), our model is equivalent to the non-negative matrix factorization (NMF) [199]. However, such settings are not suitable for dealing with time-series data.
Algorithm 1. Temporal Subspace Clustering (TSC)

**Input:** Time series data $X$, $k = 0$, step size $\eta$, number of clusters $k$, parameters $s, \lambda_1, \lambda_2, \alpha$

**Output:** Clustering index vector $Y$

1: Construct matrices $W$, $\tilde{D}$, and $L$ according to Section 5.2.1;
2: **while** not converged **do**
3: Update $V_{(k+1)}$ using (5.10), given others fixed;
4: Update $U_{(k+1)}$ using (5.13), given others fixed;
5: Update $Z_{(k+1)}$ using (5.14), given others fixed;
6: Update $D_{(k+1)}$ using (5.15), given others fixed;
7: Update $\Pi_{k+1}$: $\Pi_{k+1} = \Pi_k + \eta \alpha (V_{k+1} - Z_{k+1})$;
8: Update $\Lambda_{k+1}$: $\Lambda_{k+1} = \Lambda_k + \eta \alpha (U_{k+1} - D_{k+1})$;
9: $k = k + 1$
10: **end while**
11: Build an undirected graph $G$ using (5.17);
12: Use NCut to generate $k$ clusters, get index $Y$.

In Algorithm 1, we initialize $D$ and $Z$ with random values. All the other variables such as $U$ and $V$ are initialized with zero. To evaluate the efficiency of our algorithm, we present the analysis of time complexity. The most time-consuming step in Algorithm 1 is step 3, which costs $O(r^2 n)$. Let $t$ denote the number of iterations, the overall computational complexity of our algorithm is $O(t n r^2)$, which enjoys a good scalability w.r.t. the sample size $n$.

5.3 Experiments

In this section, we compare our approach with several state-of-the-art subspace clustering approach on three human action and gesture datasets.

5.3.1 Settings

In the experiments we utilize three public datasets, including the Keck dataset [200], Weizmann dataset [201], and Multi-Modal Action Detection (MAD) dataset [202].

**Baselines.** Our TSC approach is compared with the following representative clustering methods:

- Sparse Subspace Clustering (SSC) [14], which enforces a sparse constraint on the representation coefficients.
Table 5.1: Clustering accuracies with standard derivation and running time of all compared methods on Keck dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy (%)</th>
<th>NMI</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSC [14]</td>
<td>26.81 ± 2.41</td>
<td>0.2861</td>
<td>59.05</td>
</tr>
<tr>
<td>LRR [13]</td>
<td>12.84 ± 3.75</td>
<td>0.0617</td>
<td>14.82</td>
</tr>
<tr>
<td>LSR [16]</td>
<td>38.22 ± 2.09</td>
<td>0.3244</td>
<td>6.89</td>
</tr>
<tr>
<td>OSC [183]</td>
<td>41.89 ± 2.30</td>
<td>0.4933</td>
<td>461.18</td>
</tr>
<tr>
<td>TSC (Ours)</td>
<td><strong>57.12 ± 2.13</strong></td>
<td><strong>0.6695</strong></td>
<td><strong>49.05</strong></td>
</tr>
</tbody>
</table>

- Low-Rank Representation (LRR) [13], which incorporates a low-rank constraint on the coefficients.
- Least Square Regression (LSR) [16], which adopts a regression based formulation for subspace clustering. It’s a special case of our TSC method when $D$ is fixed, $\lambda_2 = 0$, and ignoring the non-negative constraint.
- Ordered Subspace Clustering (OSC) [183], which explicitly enforces the consecutive columns of $Z$ to be similar. It achieves the state-of-the-art results for clustering sequential data.

For those compared methods, we use the codes provided by the authors, and fine tune the parameters to achieve the best performance. Further, we will discuss how to choose parameters of our approach in the next subsections.

In the experiments, we use the clustering accuracy (AC) and normalized mutual information (NMI) as the evaluation metrics.
5.3.2 Results on Gesture Dataset

The Keck gesture data consists of 14 different gestures [200], which originally come from military signals. Each gesture is performed by three subjects, so there are three sequences for each gesture. In each sequence, the same gesture is repeated three times. Figure 5.2 shows the 14 gestures of one subject in the dataset. The original resolution of each frame is $480 \times 640$. To speed up the computation, we downsample each frame to the size of $80 \times 106$. Following [190], we extract binary masks and compute the Euclidean distance transform as frame-level features. Then we build a dictionary of temporal words with 100 clusters using the $k$-means clustering, and encode each frame as a 100 dimensional binary vector.

We concatenate the 14 gesture video sequences of each subject into a single long video sequence, and evaluate the performance of different methods. We also evaluate the computational
cost of different methods. The machine used in our experiments installs 24 GB RAM and Intel Xeon W3350 CPU. The parameters $s$, $\lambda_1$ and $\lambda_2$ are empirically set to 6, 0.01 and 15, respectively. Table 5.1 reports the average clustering accuracy and the running time. Our TSC approach significantly outperforms the other compared methods, and improves the average accuracy by at least 15%.

Figure 5.3 shows the details of clustering results of one random experiment, by rendering clusters as different colors. It shows that SSC, LRR and LSR can not obtain meaningful temporal segments, as they do not consider the temporal information. OSC and our TSC methods can obtain continuous segments in most cases. Moreover, because of the temporal Laplacian regularization function and the expressive dictionary, our approach is able to correctly recover the subspace structures in temporal space, and therefore achieves clearer sequential subspace structures than OSC.

In addition, we notice an interesting phenomenon from the clustering results of TSC in Figure 5.3. It shows that each cluster contains a short blue sequence at the beginning. After looking into the video sequences, we find that, at the beginning of each sequence, the subject walked towards the center of the room, and then performed the required gestures. It demonstrates our approach has the ability to discover undefined clusters, which might be important in some high-level vision tasks, such as video understanding.

5.3.3 Results on Action Datasets

We evaluate the action clustering performance of our approach and compared methods on the Weizmann dataset [201] and the MOD dataset [202]. The action data in the Weizmann dataset are organized in isolated clips, which provides an ideal controlled evaluation platform for temporal clustering. In addition, the MAD dataset contains continuous actions, and the start and end of each
action is provided. It provides a more realistic scenario for temporal clustering.

**Weizmann dataset.** The Weizmann dataset contains 90 video sequences (180 × 144 pixels, 50fps) captured from 9 subjects [201]. Each subject performs 10 different actions, including jumping-jack (or shortly jack), jump-forward-on-two-legs (jump), jump-in-place-on-two-legs (pjump), gallop-sideways (side), bend, skip, walk, run, wave-one-hand (wave1), and wave-two-hands (wave2). Figure 5.4 shows 10 frames of different actions in the dataset. Following the settings in [201], we extract binary masks and compute the Euclidean distance transform as frame-level features, and utilize the bag-of-words model to encode the features as binary vectors.

In this dataset, each video sequence only contains a single action. To evaluate the clustering performance of our approach and related methods, we follow the experimental protocol in [190], and concatenate multiple single-action sequences into a longer video sequence. In particular, we
CHAPTER 5. ROBUST DICTIONARY LEARNING FOR HUMAN MOTION SEGMENTATION

Table 5.2: Clustering accuracies (with standard derivation) and running time of all compared methods on Weizmann dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy (%)</th>
<th>NMI</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSC [14]</td>
<td>38.81±3.28</td>
<td>0.1214</td>
<td>289.53</td>
</tr>
<tr>
<td>LRR [13]</td>
<td>43.55±3.75</td>
<td>0.1365</td>
<td>10.26</td>
</tr>
<tr>
<td>LSR [16]</td>
<td>40.11±2.94</td>
<td>0.1164</td>
<td>3.61</td>
</tr>
<tr>
<td>OSC [183]</td>
<td>65.89±3.27</td>
<td>0.4655</td>
<td>692.14</td>
</tr>
<tr>
<td>TSC (Ours)</td>
<td>76.15±2.88</td>
<td>0.6844</td>
<td>34.16</td>
</tr>
</tbody>
</table>

Table 5.3: Clustering accuracies (%) with standard derivation of all compared methods on MAD dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Average±Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSC [14]</td>
<td>34.78</td>
<td>39.25</td>
<td>38.71</td>
<td>38.24</td>
<td>30.05</td>
<td>36.21±3.86</td>
</tr>
<tr>
<td>LRR [13]</td>
<td>35.30</td>
<td>39.82</td>
<td>40.15</td>
<td>39.71</td>
<td>31.23</td>
<td>37.24±3.91</td>
</tr>
<tr>
<td>LSR [16]</td>
<td>36.12</td>
<td>40.73</td>
<td>38.54</td>
<td>40.10</td>
<td>33.15</td>
<td>37.73±3.11</td>
</tr>
<tr>
<td>OSC [183]</td>
<td>38.55</td>
<td>41.98</td>
<td>40.12</td>
<td>42.25</td>
<td>38.22</td>
<td>40.22±1.87</td>
</tr>
<tr>
<td>TSC (Ours)</td>
<td>45.92</td>
<td>50.61</td>
<td>48.14</td>
<td>49.17</td>
<td>44.52</td>
<td>47.67±2.45</td>
</tr>
</tbody>
</table>

randomly select 5 action sequences from each subject, and concatenate these sequences into a long sequence. We repeat this procedure with 10 runs. The parameters $s$, $\lambda_1$ and $\lambda_2$ are empirically set to 6, 0.001 and 15, respectively. Table 5.2 lists the average clustering accuracy (with standard derivation) and running time of each method. We can observe that our approach obtains much better results than the compared methods. The average clustering accuracy is improved by at least 10%, comparing with the state-of-the-art method OSC.

To take the close look at the clustering results, Figure 5.5 shows the details of clustering results of one random experiment, by rendering clusters as different colors. It shows that SSC, LRR and LSR can not obtain meaningful temporal segments, as they do not consider the temporal information. OSC and our TSC methods can obtain continuous segments in most cases. Moreover, because of the temporal Laplacian regularization function and the expressive dictionary, our approach achieves more clearer sequential subspace structures than OSC.

Multi-modal Action Detection (MAD) Dataset. The sequences in the Keck dataset and Weizmann dataset are isolated clips. However, manually concatenating the isolated clips results in discontinuous time series that are not valid in realistic scenario. The recently published MAD dataset contains multiple continuous actions, which is more challenging than the Weizmann dataset. The MAD dataset contains 40 sequences captured from 20 subjects (2 sequences per subject). Each subject performs all the 35 activities continuously, and the segments between two actions are
CHAPTER 5. ROBUST DICTIONARY LEARNING FOR HUMAN MOTION SEGMENTATION

Table 5.4: Normalized mutual information (NMI) of all compared methods on MAD dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Subject 1</th>
<th>Subject 2</th>
<th>Subject 3</th>
<th>Subject 4</th>
<th>Subject 5</th>
<th>Average±Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSC [14]</td>
<td>0.2241</td>
<td>0.2385</td>
<td>0.2301</td>
<td>0.2297</td>
<td>0.2154</td>
<td>0.2276±0.0085</td>
</tr>
<tr>
<td>LRR [13]</td>
<td>0.2016</td>
<td>0.2187</td>
<td>0.2248</td>
<td>0.2045</td>
<td>0.1985</td>
<td>0.2096±0.0115</td>
</tr>
<tr>
<td>LSR [16]</td>
<td>0.2401</td>
<td>0.2398</td>
<td>0.2515</td>
<td>0.2349</td>
<td>0.2207</td>
<td>0.2374±0.0111</td>
</tr>
<tr>
<td>OSC [183]</td>
<td>0.2618</td>
<td>0.2714</td>
<td>0.2703</td>
<td>0.2925</td>
<td>0.2544</td>
<td>0.2701±0.0143</td>
</tr>
<tr>
<td>TSC (Ours)</td>
<td><strong>0.3435</strong></td>
<td><strong>0.3677</strong></td>
<td><strong>0.3520</strong></td>
<td><strong>0.3312</strong></td>
<td><strong>0.3287</strong></td>
<td><strong>0.3446±0.0160</strong></td>
</tr>
</tbody>
</table>

Figure 5.6: Example frames of MAD dataset.

considered the null class (i.e., the subject is standing) [202]. The 35 actions include full-body motion (e.g., Running, Crouching, jumping), lower-body motion (e.g., kicking), and upper-body motion (e.g., Throw, Basketball Dribble, Baseball swing). The length of each sequence is around 2-4 minutes (4000-7000 frames). Each sequence has three different modalities: RGB video (240×320), 3D depth (240×320), and a body-joint sequence (3D coordinates of 20 joints per frame). Figure 5.6 shows some frames in the MAD dataset.

We use depth sequences to generate binary masks of human, and compute the Euclidean distance transform as frame-level features. Then we build a dictionary of temporal words with 100 clusters using the $k$-means clustering, and encode each frame as a 100 dimensional binary vector. We randomly choose 5 subjects, and evaluate the clustering performance of each compared method. The parameters are fine tuned to achieve the best result of each method. Figure 5.7 shows the clustering results on MAD dataset (Subject 1) with different number of actions. It shows that our TSC approach consistently achieves much better results than other methods. Table 5.3 and Table 5.4 list the clustering accuracy and NMI for each subject, which demonstrates the effectiveness of our approach.

94
CHAPTER 5. ROBUST DICTIONARY LEARNING FOR HUMAN MOTION SEGMENTATION

Figure 5.7: Clustering results on MAD dataset (Subject 1) with different number of actions. (Left: Accuracy. Right: NMI)

Figure 5.8: Visualization of graphs learned by SSC, LRR, LSR, OSC and our TSC approach on Keck gesture dataset. The red color denotes large graph weights, while the blue color indicates small weights.
Table 5.5: Clustering accuracies of TSC and the state-of-the-art motion segmentation methods on three datasets.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Keck</th>
<th>Weizmann</th>
<th>MAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMTC [190]</td>
<td>N/A</td>
<td>68.05</td>
<td>N/A</td>
</tr>
<tr>
<td>ACA [189]</td>
<td>51.76</td>
<td>75.06</td>
<td>42.05</td>
</tr>
<tr>
<td>HACA [189]</td>
<td>53.42</td>
<td>75.80</td>
<td>45.31</td>
</tr>
<tr>
<td>TSC (Ours)</td>
<td>57.12</td>
<td>76.15</td>
<td>47.67</td>
</tr>
</tbody>
</table>

Figure 5.9: Sensitivity analysis on Weizmann dataset. Left: accuracy with different values of parameters; Right: accuracy with different number of clusters.

5.3.4 Discussions

Graph Visualization. Constructing an effective graph is the key in clustering methods. Indeed, existing subspace clustering methods and our approach mainly focus on estimating the coding matrix for graph construction. To illustrate why our approach performs much better than its competitors, we visualize the graphs learned by SSC, LRR, LSR, OSC and our approach in Figure 5.8. By considering the sequential relationships in time-series data, we can observe the much denser block diagonals in the graphs of OSC and our approach compared to other graphs. It implies that the within-cluster structures are enhanced in OSC and our graph. Moreover, as our approach is more flexible to control the sequential neighbors, the graph structure of our approach is clearer than OSC.

Comparisons with Motion Segmentation Methods. We also compare the clustering performance between our method and the motion segmentation methods. MMTC is a maximum margin clustering method. ACA is based on spectral clustering, and HACA is an improved version of ACA. To the best of our knowledge, HACA is the state-of-the-art unsupervised motion analysis method. Table 5.5
shows the clustering results on three datasets. It shows that HACA usually performs better than ACA and MMTC, while our TSC method consistently outperforms others.

**Parameter Sensitivity.** There are two major parameters in our model, $\lambda_1$ and $\lambda_2$. Figure 5.9(Left) shows the clustering accuracy of our TSC approach with different values of $\lambda_1$ and $\lambda_2$. We can observe clustering results are not very sensitive to $\lambda_1$ in the range $[0, 0.002]$. Meanwhile, $\lambda_2 = 10$ can lead to the best clustering result. We can also validate the effectiveness of the temporal Laplacian regularization function from Figure 5.9 (Left). In practice, we may not know the true number of clusters. For sensitivity analysis, we vary the desired number of clusters but fix the number of true classes. In this case, the evaluation metrics like accuracy and NMI cannot be directly applied, since there was no one-to-one mapping between the generated clusters and ground truth. Instead, we use a *pair-counting* measurement designed in [190]. Consider all pairs of same class video frames, $p_1$ is defined as the percentage of pairs of which both frames were assigned to the same cluster. Consider all pairs of different-class video frames, $p_2$ is defined as the percentage of pairs of which two frames were assigned to different clusters. Moreover, $p_3$ is the average of $p_1$ and $p_2$, which shows the clustering performance. Figure 5.9 (Right) shows the values of $p_1$, $p_2$ and $p_3$ by varying the desired number of clusters from 2 to 7. We observe that the summarized value $p_3$ is relatively stable in a wide range.

In addition, Figure 5.10 shows the clustering results with different size of dictionary and different number of sequential neighbors on the Keck dataset. It shows that our approach is not very sensitive to the size of dictionary in a wide range. Clustering accuracy would decrease slightly when the number of sequential number increases.
5.4 Summary

We propose a temporal subspace clustering (TSC) approach in this chapter. TSC considers the sequential information in time-series data by virtue of a temporal Laplacian regularization term. In addition, a non-negative dictionary is learned to form an expressive encoding space. We design an efficient ADMM optimization algorithm to solve the problem. Experimental results on human action and gesture datasets show that TSC significantly outperforms the state-of-the-art subspace clustering methods. Specifically, our TSC approach improves the average clustering accuracy by at least 10%. In our future work, we will design algorithms to automatically find the number of clusters for time-series data. We would also apply the proposed temporal Laplacian regularization function to other temporal analysis tasks.
Chapter 6

Robust Dictionary Learning for Knowledge Transfer

6.1 Background and Motivation

The performance of visual learning algorithms is heavily dependent on the choice of data representation [203]. Sparse coding [204, 95], dictionary learning [147, 38] and low-rank learning [13, 181, 171, 205] have been widely used for representing visual data. Good representations are expressive, meaning that a reasonably sized dictionary (basis functions) can capture a huge number of possible input configurations, and also characterize a given set of data with certain global structural blueprint (i.e. multiple clusters, subspaces, or manifolds). However, the lack of training data presents a common challenge in many sophisticated representation learning algorithms.

Traditionally, this problem was partially tackled by semi-supervised learning [53] or transfer learning methods [54, 132, 131, 206]. Semi-supervised learning makes use of the labeled sample set and a larger set of unlabeled samples, which are drawn from the same domain with same distribution, to train a model. In other words, semi-supervised learning can only solve learning problems in the same domain. In transfer learning, this restriction is relaxed to some extent. The labeled samples and auxiliary samples in transfer learning are drawn from different domains with different distributions. But transfer learning requires that two domains should be similar to each other. Most transfer learning methods assume that two domains share a similar knowledge structure that defines the domain relatedness. In a word, both semi-supervised learning and transfer learning usually put strong restrictions on auxiliary (source) data, which limited their applicability. Recently,
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

an emerging machine learning topic of self-taught learning (STL) [55, 56, 57, 58, 59, 207, 60] using unlabeled data with fewer restrictions holds significant promise in terms of enhancing the performance of image clustering and classification. Raina et al. first proposed the concept of self-taught learning by applying sparse coding mechanism to construct a higher-level representation from the unlabeled data [55, 58]. Lee et al. extended Raina’s work by presenting a generalization of sparse coding module which could be suited to model other data types drawn from any exponential family distribution [56]. From the application point of view, Dai et al. proposed a clustering algorithm in the spirit of self-taught learning by allowing the feature representation from the auxiliary data to influence the target data through a common set of features [57]. Kuen et al. employed the core idea of self-taught learning, and transferred stacked auto encoders for visual tracking [60]. However, existing self-taught learning methods do not take advantage of any global structure information in the target set, as they encode each input signal independently. Besides, a generalizable schema of self-taught learning for both supervised and unsupervised learning tasks has not been well studied yet.

Self-taught learning and transfer learning are two related concepts [54]. The key difference is that they place different restrictions on the auxiliary domain. In particular, transfer learning only leverages labeled data from related homogenous tasks (e.g., domain adaptation [208]), while self-taught learning relaxes such restriction by utilizing arbitrary images (e.g., randomly downloaded images) to form the auxiliary domain. The intuition behind self-taught learning is that randomly selected visual data in an auxiliary domain can still contain the basic visual patterns (such as edges, corners, atomic shapes) that are similar to those in the target domain. The flexibility of self-taught learning makes it particularly potential to ever-increasing huge amount of unlabeled visual data. Existing self-taught learning methods, however, simply ignore the structure information in the target domain, which is critical in the visual learning tasks such as image classification.

In this chapter, we propose a novel self-taught low-rank (S-Low) coding framework for visual learning [209, 210]. By leveraging a high quality dictionary abstracted from the wealth of information behind the auxiliary domain, we aim to learn expressive high-level representations for the target domain. Since many types of visual data are well characterized by subspace structure [24, 18], we introduce a low-rank constraint in our framework to take advantage of global structure information in the target domain. Emphasizing such kind of structure information through low-rank constraints in our approach could greatly benefit broad visual learning tasks. In particular, our approach is very suitable for addressing the tasks that leverage on the exploitation of underlying data structure, such as object recognition, scene classification, face recognition, image clustering, etc. Especially when
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

Figure 6.1: Diagram of the self-taught low-rank (S-Low) coding framework. A small target dataset $X_T$ is usually not sufficient to extract effective features. By utilizing the auxiliary dataset $X_S$, the proposed S-Low framework learns a shared dictionary $D$ from two domains, and enforces a low-rank constraint on the coefficient matrix of target domain $Z_T$ that is considered as new feature representations. Finally, the normalized cut (NCut) algorithm can be utilized for image clustering, and the support vector machine (SVM) can be trained on $Z_T$ for image classification.

The target data set is small, our approach is still able to extract effective feature representations by virtue of large-scale unlabeled data in the auxiliary domain. The low-rank constraint is also capable of removing noise and outliers from data [23, 211], which helps us learn more robust representations in the target domain.

Fig. 6.1 illustrates the diagram of our approach. Intuitively, we extract useful building blocks from auxiliary domain in terms of a good characterization of underlying structure in the target domain. An expressive dictionary is learned by modeling both auxiliary domain and target domain. In this process, the structure information in target domain is enforced using low-rank constraints. More specifically, our approach can be formulated as a rank-minimization and dictionary learning problem, and we design an effective majorization-minimization optimization algorithm to jointly learn the dictionary and low-rank codings. Finally, the learned low-rank codings correspond to the target domain can be directly used for clustering, or can be employed to train a supervised model like support vector machines (SVM) for classification.

Moreover, some limitations of existing self-taught learning methods can be addressed by the proposed method. First, existing methods always either loosely combine representation learning and final visual tasks [55], or tailor the algorithms to particular applications [59]. Our approach could be easily applied to both supervised and unsupervised learning tasks in a general way. Second, existing self-taught methods learn new representations independently for each sample in the target
domain, where the important global structural information in the given set is simply ignored. Our approach could effectively utilize the rich low-level pattern information abstracted from the auxiliary domain to intelligently characterize the high-level structure information in the target domain. It closely link the coding procedure to the learning tasks.

Our work is closely related to two topics, including self-taught learning and low-rank modeling. The most relevant method in the literature is the robust and discriminative self-taught learning (RDSTL) [59]. RDSTL is a classification algorithm with self-taught nature by utilizing supervision information in the target domain to discover the optimal dictionary basis vectors. There are significant differences between RDSTL and our approach. First, RDSTL does not consider the global structure information in target domain, which is carefully modeled via low-rank constraints in our approach. Second, the $l_{2,1}$ norm used in RDSTL is a biased estimator, while our approach employs the unbiased estimators including matrix $\gamma$-norm and minimax concave penalty (MCP) norm. Third, RDSTL is designed for classification. We present both clustering and classification algorithms using our framework.

Some recent works introduced low-rank constraints into transfer learning problems [132, 131, 212]. Low-rank transfer subspace learning method imposes a low-rank constraint on a low-dimensional subspace shared by source and target domains [132], and low-rank domain adaptation method aims to reduce the domain distribution disparity using low-rank representations [131]. A latent low-rank transfer learning approach is proposed to tackle the missing modality recognition problem [213]. Most recently, low-rank constraints are incorporated into deep learning architecture to achieve transfer learning [214]. Our approach differs from them in three aspects. First, these methods have strong restrictions in terms of using related homogenous tasks in source and target domains, while our approach relaxes such restrictions. Second, they cannot learn dictionaries due to their problem settings. Third, the knowledge across different domains are transferred via a shared subspace, while our approach transfers knowledge via a dictionary.

In summary, the major contributions of this chapter include the following:

1. With the help of rich information from the auxiliary domain, we learn effective feature representations, S-Low codings, by incorporating low-rank constraints in the target domain.

2. The proposed self-taught learning approach is a general framework, which can be applied to various visual learning scenarios. In the chapter, we present detailed algorithms for unsupervised learning and supervised learning.

3. Instead of using the biased estimators like nuclear norm and $l_1$ norm in many existing low-rank matrix recovery methods, we relax the matrix rank and $l_0$ norm in our model by two unbiased
estimators, matrix $\gamma$-norm and minimax concave penalty (MCP) norm. An effective majorization-minimization optimization algorithm is developed to solve our model. We also empirically illustrate the convergence property of the optimization algorithm.

(4). Extensive experimental results on five benchmark datasets demonstrate that our approach consistently outperforms several representative low-rank learning and self-taught learning methods.

6.2 Self-Taught Low-Rank (S-Low) Coding

In this section, we formulate the proposed self-taught low-rank coding mechanism, and develop our approach systematically. Then we present an effective optimization algorithm to solve the model. Table 6.1 summarizes the notations used throughout this chapter.

6.2.1 Motivation

Our goal is to take advantages of the abundant unlabeled data, in order to improve the coding performance for various visual learning tasks. To achieve this goal, we propose a self-taught low-rank (S-Low) coding framework, by leveraging a high quality dictionary abstracted from the wealth of information behind the auxiliary domain. Our intuition is that many types of visual data are well characterized by subspace structure, and therefore it is possible to leverage on such information from both auxiliary and target domains, and finally learn expressive high-level representations for the target domain. Specifically, we introduce a low-rank constraint in our framework to take advantage of global structure information in the target domain. Emphasizing such kind of structure information through low-rank constraints could greatly benefit broad visual learning tasks especially clustering and classification, in which recognizing the underlying structure of a given sample set is our ultimate goal. The low-rank constraint is also capable of removing noise and outliers from data [23, 211], which helps us learn more robust representations in the target domain.

6.2.2 Problem Formulation

Considering the self-taught learning problem, we are given a set of abundant, unlabeled samples, $X_S = \{x_{S1}, \cdots, x_{Sm}\} \in \mathbb{R}^{d \times m}$, in the auxiliary domain (or source domain), and we also have limited samples in the target domain, $X_T = \{x_{T1}, \cdots, x_{Tn}\} \in \mathbb{R}^{d \times n}$. Our approach aims to learn expressive codings, in which the subspace structural information is encoded, for the samples
Table 6.1: Notations

<table>
<thead>
<tr>
<th>Notations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_S \in \mathbb{R}^{d \times m}$</td>
<td>Unlabeled samples in auxiliary domain</td>
</tr>
<tr>
<td>$X_T \in \mathbb{R}^{d \times n}$</td>
<td>Samples in target domain</td>
</tr>
<tr>
<td>$D \in \mathbb{R}^{d \times r}$</td>
<td>Dictionary</td>
</tr>
<tr>
<td>$Z_S \in \mathbb{R}^{r \times m}$</td>
<td>Low-rank codings for auxiliary samples</td>
</tr>
<tr>
<td>$Z_T \in \mathbb{R}^{r \times n}$</td>
<td>Low-rank codings for target samples</td>
</tr>
<tr>
<td>$E_S \in \mathbb{R}^{d \times m}$</td>
<td>Sparse noise in auxiliary samples</td>
</tr>
<tr>
<td>$E_T \in \mathbb{R}^{d \times n}$</td>
<td>Sparse noise in target samples</td>
</tr>
<tr>
<td>$|\cdot|_\gamma$</td>
<td>Matrix $\gamma$-norm</td>
</tr>
<tr>
<td>$M_{\lambda,\gamma}(\cdot)$</td>
<td>Matrix concave penalty norm</td>
</tr>
<tr>
<td>$d$</td>
<td>Dimensionality of each sample</td>
</tr>
<tr>
<td>$m$</td>
<td>Number of auxiliary samples</td>
</tr>
<tr>
<td>$n$</td>
<td>Number of target samples</td>
</tr>
<tr>
<td>$r$</td>
<td>Size of dictionary</td>
</tr>
</tbody>
</table>

in the target domain. Like other self-taught learning methods, we do not assume that the data from auxiliary and target domains share the same (or similar) distributions. Furthermore, we do not require that the samples are labeled in the target domain. Therefore, our approach can be performed in either unsupervised or supervised fashions, which differs from the problem settings in [55] and [59]. We will show that our approach could deal with clustering problem if labels are unavailable in the target domain, or classification problem with labeled samples.

Traditionally, the sparse coding [204, 95], dictionary learning [147, 38] or low-rank learning [13, 181] methods approximately represent the samples in a single domain (i.e., the target domain):

$$X_T \approx D_T Z_T,$$  \hspace{1cm} (6.1)

where $Z_T \in \mathbb{R}^{r \times n}$ is the representation coefficient matrix and $D_T \in \mathbb{R}^{d \times r}$ is a dictionary. $r$ is the size of dictionary. Here $Z_T$ is usually expected to be sparse or low-rank, according to the application scenario. Note that the dictionary $D_T$ is often set as the sample set in some sparse representation and low-rank learning methods [95, 13, 24, 181] (i.e., $D_T = X_T$), which may suffer the insufficient sampling problem.

With the help of auxiliary domain, we are able to learn a more informative dictionary, and also tackle the insufficient data sampling problem.

**First**, we can learn the dictionary from all the available samples in two domains. The whole sample set is $X = [X_S \ X_T]$. We aim to represent all samples in $X$ using a dictionary $D \in \mathbb{R}^{d \times r}$. Therefore, we introduce the constraint $[X_S \ X_T] = D[Z_S \ Z_T] + [E_S \ E_T]$, where
$Z_S \in \mathbb{R}^{r \times m}$ and $Z_T \in \mathbb{R}^{r \times n}$ are the coefficient matrices corresponding to auxiliary domain and target domain, respectively. $E_S$ and $E_T$ are the sparse noise matrices that model the reconstruction errors in auxiliary and target domains. The noise matrices $E_S \in \mathbb{R}^{d \times m}$ and $E_T \in \mathbb{R}^{d \times n}$ are often constrained using the surrogate of $l_0$ norm, such as $l_1$ or $l_{2,1}$ norms. In reality, target samples may contain various types of noise. Considering the sparse noise matrices in the model enables us to learn a robust dictionary.

**Second,** for many vision problems like clustering or classification, samples in the target domain usually lie in several underlying subspaces. Many recent research efforts [24, 29, 181] have shown that enforcing low-rank constraint is an effective way to discover those underlying subspace structure. Leveraging such structure information can greatly benefit the visual learning tasks. In light of this observation, we impose a low-rank constraint on the coefficient matrix $Z_T$ in the target domain, where the learning tasks are performed. Then, our objective function is formulated as follows:

\[
\min_{D, Z_S, Z_T, E_S, E_T} \text{rank}(Z_T) + \lambda_1 \|E_S\|_0 + \lambda_2 \|E_T\|_0,
\]

subject to

\[
X_S = DZ_S + E_S, \quad X_T = DZ_T + E_T,
\] (6.2)

where $\text{rank}(\cdot)$ denotes the rank function, $\|\cdot\|_0$ is the $l_0$ norm, $\lambda_1$ and $\lambda_2$ are two trade-off parameters to balance the effects of different terms.

The first term in (6.2) characterizes the low-rankness of $Z_T$ in the target domain, and the last two terms model the reconstruction errors. Eq.(6.2) is a variant of rank minimization problem that is NP-hard in general. Therefore, it cannot be solved directly. In practice, the rank function and $l_0$ norm can be relaxed by the nuclear norm and $l_1$ norm, respectively. Some convex optimization tools, such as inexact augment Lagrange multiplier (ALM) algorithm, can achieve acceptable performance. However, it has been noted that the nuclear norm and $l_1$ norm are biased estimators, as they over penalize large singular values and large entries [215]. To tackle this problem, we employ the non-convex surrogates of rank function and $l_0$ norm, which are matrix $\gamma$-norm and minimax concave penalty (MCP) norm, respectively.

The matrix $\gamma$-norm for a matrix $A \in \mathbb{R}^{p \times q}$ is defined as [215]:

\[
\|A\|_\gamma = \sum_{i=1}^{s} \frac{\sigma_i(A)}{\gamma + \left(1 - \frac{u_i}{s}\right)}
\]

\[
= \sum_{i=1}^{s} \phi_{1,\gamma}(\sigma_i(A)) = M_\gamma(\sigma(A)), \quad \gamma > 1,
\] (6.3)
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

where \( \sigma(A) = (\sigma_1(A), \cdots, \sigma_s(A))^T \) denotes a function from \( \mathbb{R}^{p \times q} \) to \( \mathbb{R}^s \), \( s = \min(p, q) \). The matrix \( \gamma \)-norm is non-convex w.r.t \( A \).

The matrix MCP norm is defined as [216]:

\[
M_{\lambda, \gamma}(A) = \sum_{i,j} \phi_{\lambda, \gamma}(A_{i,j}), \tag{6.4}
\]

where

\[
\phi_{\lambda, \gamma}(t) = \lambda \int_0^t [1 - \frac{x}{\gamma \lambda}]_+ dx = \begin{cases} 
\gamma \lambda^2 / 2, & \text{if } |t| \geq \gamma \lambda \\
\gamma |t| - \frac{t^2}{2 \gamma}, & \text{otherwise}.
\end{cases}
\]

\([z]_+ = \max(z, 0)\). Here, we choose \( \lambda = 1 \), and denote \( M_{\gamma}(A) = M_{1, \gamma}(A) \) for simplicity.

By replacing the rank function and \( l_0 \) norm with matrix \( \gamma \)-norm and MCP norm, the objective function (6.2) can be rewritten as:

\[
\min_{D, Z_S, Z_T, E_S, E_T} \|Z_T\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E_S) + \lambda_2 M_{\gamma_2}(E_T), \tag{6.5}
\]

\[s.t. \quad X_S = DZ_S + E_S, X_T = DZ_T + E_T.\]

Third, the dictionary is jointly learned from both auxiliary and target domains, in order to transfer useful knowledge from the auxiliary domain. The two constraints in (6.5) share the same dictionary \( D \). As the source dataset \( X_S \) usually contains much more samples than target dataset \( X_T \), the learning of dictionary is easily dominated by the source data. However, it is more rational to emphasize the reconstruction power of \( D \) in the target domain in which our learning task performs. Therefore, we introduce an \( l_{2,1} \) norm constraint on the source coefficient matrix \( Z_S \). In this way, some rows in \( Z_S \) are encouraged to be zero, which enables \( X_S \) to adaptively select bases from \( D \). On the other hand, \( D \) is fully used to reconstruct samples in the target domain.

After that, our objective becomes:

\[
\min_{D, Z_S, Z_T, E_S, E_T} \|Z_T\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E_S) + \lambda_2 M_{\gamma_2}(E_T) + \lambda_3 \|Z_S\|_{2,1} \tag{6.6}
\]

\[s.t. \quad X_S = DZ_S + E_S, X_T = DZ_T + E_T,\]

where \( \lambda_3 \) is a trade-off parameter, and \( \|Z_S\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{d} ([Z_S]_{ij})^2} \) is the \( l_{2,1} \) norm.

Each column in the learned coefficient matrix \( Z_T \) corresponds to one sample in the target domain, which is named low-rank coding in our chapter.
6.2.3 Optimization

In this section, we design a MM-ALM algorithm to solve (6.6). We first introduce the generalized singular value shrinkage operator $S_{\tau, \Lambda}$ and generalized shrinkage operator $D_{\gamma, W}$ [215]:

$$S_{\tau, \Lambda} = U_X D_{\tau, \Lambda}(\Sigma_X)(V_X)^T,$$

(6.7)

$$[D_{\gamma, W}(A)]_{ij} = \text{sgn}(A_{ij})(|A_{ij} - \tau W_{ij}|)_+,$$

(6.8)

where $\Sigma$ and $\Lambda$ are non-negative matrices.

To facilitate the optimization, we add a relaxation variable $J \in \mathbb{R}^{r \times n}$ to (6.6):

$$\min_{D, Z, S, Z^T, E_S, E_T, J} \|J\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E_S) + \lambda_2 M_{\gamma_2}(E_T) + \lambda_3 \|Z_S\|_{2,1},$$

(6.9)

s.t. $X_S = DZ_S + E_S, X_T = DZ_T + E_T, Z_T = J.$

The MM-ALM algorithm consists of an outer loop and an inner loop. In each iteration, the outer loop replaces the non-convex problem by its locally linear approximation (LLA) to form a weighted convex problem, while an inner loop is an inexact ALM algorithm.

In the outer loop, we reformulate the objective function as follows. Since the objective function in Eq. (6.9) is concave w.r.t. $(\sigma(J), |E_S|, |E_T|)$, we can approximate $\|J\|_{\gamma_1} + \lambda_1 M_{\gamma_2}(E_S) + \lambda_2 M_{\gamma_2}(E_T)$ by its LLA at $(\sigma(J)^{old}, |E_S|^{old}, |E_T|^{old})$, and we obtain the following objective function:

$$\min_{D, Z_S, Z_T, E_S, E_T, J} Q_{\gamma_1}(\sigma(J)|\sigma(J)^{old}) + \lambda_1 Q_{\gamma_2}(E_S|E_S^{old}) + \lambda_2 Q_{\gamma_2}(E_T|E_T^{old}) + \lambda_3 \|Z_S\|_{2,1},$$

(6.10)

s.t. $X_S = DZ_S + E_S, X_T = DZ_T + E_T, Z_T = J,$

where

$$Q_{\gamma}(A|A^{old}) = M_{\gamma}(A^{old}) + \sum_{i,j} (1 - |A_{ij}^{old}|/\gamma)_+ (|A_{ij}| + |A_{ij}^{old}|).$$

is the LLA of $M_{\gamma}(A)$ given $A^{old}$.

In the inner loop, we utilize the inexact ALM algorithm to solve Eq. (6.10) by alternately updating different sets of variables.
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

Given an initialized dictionary \( D \), we update other variables \( J, Z_S, Z_T, E_S \) and \( E_T \). The augmented Lagrangian function of (6.10) is

\[
\mathcal{L} = Q_{\gamma_1}(\sigma(J)|\sigma(J)^{old}) + \lambda_1 Q_{\gamma_2}(E_S|E_S^{old}) \\
+ \lambda_2 Q_{\gamma_2}(E_T|E_T^{old}) + \lambda_3 \|Z_S\|_{2,1} + \text{Tr}(R^T(Z_T - J)) \\
+ \text{Tr}(Y^T(X_S - DZ_S - E_S)) \\
+ \text{Tr}(Q^T(X_T - DZ_T - E_T)) \\
+ \frac{\mu}{2}(\|X_S - DZ_S - E_S\|_F^2 + \|X_T - DZ_T - E_T\|_F^2) \\
+ \|Z_T - J\|_F^2,
\]

where \( \|.\|_F \) is the Frobenius norm, \( Y \in \mathbb{R}^{d \times m}, Q \in \mathbb{R}^{d \times n} \) and \( R \in \mathbb{R}^{r \times n} \) are Lagrange multipliers and \( \mu > 0 \) is a positive penalty parameter.

In particular, we alternately update these variables in the \( k + 1 \) iteration as follows:

\[
J_{k+1} = S_{1/\mu}(Z_{Tk} + (R_k/\mu_k)),
\]

\[
Z_{T(k+1)} = (I_n + D^T D)^{-1}(D^T X_T - D^T E_{Tk} + J_{k+1} + (D^T Q_k - R_k)/\mu_k),
\]

\[
Z_{S(k+1)} = \min_Z \frac{\lambda_k}{\mu_k} \|Z\|_{2,1} + \frac{1}{2} \|Z - Z_{Sk} - D^T(X_S - DZ_{Sk} - E_{Sk} + Y_k/\mu_k)\|_F^2,
\]

\[
E_{S(k+1)} = D_{\lambda_1/\mu,W}(X_S - DZ_{S(k+1)} + Y_k/\mu_k),
\]

\[
E_{T(k+1)} = D_{\lambda_2/\mu,W}(X_T - DZ_{T(k+1)} + Q_k/\mu_k),
\]

When the variables \( J, Z_S, Z_T, E_S \) and \( E_T \) are optimized, we update the dictionary \( D \) using an efficient solver presented in [204]. More specifically, by ignoring the irrelevant terms in (6.11), we compute \( D^{j+1} \) by minimizing the following objective:

\[
D^{j+1} = \min_D \text{Tr}(Y^T(X_S - DZ_S - E_S)) + \text{Tr}(Q^T(X_T - DZ_T - E_T)) \\
+ \frac{\mu}{2}(\|X_S - DZ_S - E_S\|_F^2 + \|X_T - DZ_T - E_T\|_F^2).
\]

6.2.4 Algorithm and Discussions

The whole optimization process, including both inner loop and outer loop, is repeated until convergence. The problem (6.14) can be solved according to the Lemma 3.2 in [13]. The detailed procedures of our optimization is outlined in Algorithm 1.

**Convergence.** Lemma 1 demonstrates the local convergence property of our algorithm.
Lemma 1. When $D$ is fixed, the objective function values of (6.10) obey

$$f(J, E_S, E_T, Z_S) \leq Q_{\gamma_1}(\sigma(J)|\sigma(J)^{old}) + \lambda_1 Q_{\gamma_2}(E_S|E_S^{old}) + \lambda_2 Q_{\gamma_2}(E_T|E_T^{old}) + \lambda_3 \|Z_S\|_{2,1}$$

$$\leq Q_{\gamma_1}(\sigma(J)^{old}|\sigma(J)^{old}) + \lambda_1 Q_{\gamma_2}(E_S^{old}|E_S^{old}) + \lambda_2 Q_{\gamma_2}(E_T^{old}|E_T^{old}) + \lambda_3 \|Z_S^{old}\|_{2,1}$$

$$= f(J^{old}, E_S^{old}, E_T^{old}, Z_S^{old}).$$

This lemma can be easily proved using the Proposition 4 in [215]. It demonstrates the local convergence property of our algorithm.

Initialization. In Algorithm 1, the dictionary $D$ is initialized by some randomly selected samples from $X$. $Z_S$ and $Z_T$ are initialized by random normal matrices. All the other variables are initialized by 0. Our experiments show that both $D$ and $Z$ are not sensitive to the random selection of variables.
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

initializations.

Time Complexity. Given \( r < n \), the step 4 in Algorithm 1 involves SVD decomposition of a matrix with size \( r \times n \) that costs \( O(nr^2) \), and the multiplication and inverse of matrices in step 5 also cost \( O(nr^2) \). Because the outer loop converges quickly in practice, which will be illustrated in experiments, we only consider the inner loop in the time complexity analysis. Let \( t \) denote the number of iterations in the inner loop, the complexity of our algorithm is \( O(tnr^2) \).

6.3 Learning with S-Low Coding

In this section, we present two learning algorithms based on our S-Low coding approach, including clustering and classification.

6.3.1 S-Low Clustering

Given an unlabeled sample set \( X = [X_S \ X_T] \) in the self-taught learning scenario, the goal of our S-Low clustering algorithm is to correctly recover the underlying subspaces in the target domain.

The low-rank codings \( Z_T \) for the target domain are utilized to define an affinity matrix of an undirected graph \( G \). According to the low-rank subspace recovery theory, each column in coefficient matrix \( Z \) could serve as a new representation for a sample, and then the correlation coefficient of each pair of samples would be a good choice for weighting the corresponding edge in the undirected graph [93]. In particular, we calculate the cosine similarity of each pair of samples (i.e., two vectors) as the graph weights. Given two coding vectors \( z_i, z_j \in Z_T \), the graph weight \( G(i, j) \) is defined as

\[
G(i, j) = \frac{z_i^T z_j}{\|z_i\|_2 \|z_j\|_2}.
\]  

(6.18)

On the other hand, sparsity is always emphasized during graph construction, and therefore we prune those edges with small weights to make the graph sparse. Finally, an effective clustering algorithm, Normalized Cuts [66], is employed to produce the clustering results. The whole procedures of S-Low clustering are summarized in Algorithm 2.

6.3.2 S-Low Classification

When label information are available in the target domain, we design a classification algorithm based on our S-Low coding approach to train a classifier. Then, with the help of the learned
Algorithm 2. S-Low Clustering Algorithm

Input: data matrix $X = [X_S \ X_T]$, nearest neighbors $K$, number of clusters $C$
1: Obtain the low-rank representation matrix $Z_T$ using Algorithm 1;
2: Build an undirected graph $G$ based on $Z_T$ (using (6.18)), where the edges are weighted using correlation coefficients of each pair of samples;
3: Prune graph $G$ by removing some edges with small weights (keep $K$ nearest neighbors for each node);
4: Use NCut to generate $C$ clusters.
Output: clustering index vector $L$

Algorithm 3. S-Low Classification Algorithm

Input: data matrix $X = [X_S \ X_T]$, class labels of $X_T$, test sample $y$
1: Obtain the low-rank representation $Z_T$ and dictionary $D$ using Algorithm 1;
2: Train a SVM classifier using $Z_T$;
3: Calculate sparse representation of $y$ using (6.19);
4: Predict class label of $y$.
Output: predicted class label $c_y$

dictionary $D$, our algorithm could classify new test samples. As discussed in Section 6.2.1, low-rank codings $Z_T$ can be regarded as new representations of the target sample set $X_T$. Given a test sample $y$, we can calculate the representation coefficients of $y \in \mathbb{R}^{d \times 1}$ by solving:

$$
\min_a \|y - Da\|_2^2 + \lambda \|a\|_1, \quad (6.19)
$$

where $a \in \mathbb{R}^{r \times 1}$ is the coefficient vector of $y$ over $D$.

Without the loss of generality, we can train any classifier using $Z_T$. In this chapter, we adopt the commonly used classifier support vector machines (SVM) [146] to predict the class label of $y$. Algorithm 3 summarizes all the procedures in our S-Low classification algorithm.

6.4 Experiments

In this section, we evaluate the performance of the proposed S-Low Coding method and corresponding learning algorithms. We first introduce the auxiliary dataset, target datasets and basic experimental settings. The convergence property and parameter sensitivity are then evaluated and
Table 6.2: Target datasets for experiments.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Categories</th>
<th># Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRC-v1</td>
<td>7</td>
<td>210</td>
</tr>
<tr>
<td>MSRC-v2</td>
<td>20</td>
<td>591</td>
</tr>
<tr>
<td>Caltech-101</td>
<td>20</td>
<td>1230</td>
</tr>
<tr>
<td>Scene-15</td>
<td>15</td>
<td>4485</td>
</tr>
<tr>
<td>Caltech-UCSD Birds</td>
<td>30</td>
<td>1622</td>
</tr>
</tbody>
</table>

Figure 6.2: Sample images in auxiliary domain (above) and target domain (below).

discussed. After that, we report and discuss the results of S-Low clustering and S-Low classification algorithms, compared with some baselines.

6.4.1 Datasets and Settings

Auxiliary Domain Dataset. Following [59], we randomly select 5000 unlabeled images from the LabelMe website1 to construct the sample set in auxiliary domain. Currently, the LabelMe dataset contains more than 100 thousand images collected from various resources, which provide us a great auxiliary domain for self-taught learning. Fig. 6.2 (a) shows some images in the LabelMe dataset.

To evaluate how the data size in the auxiliary domain affects the performance of learning tasks in the target domain, we alter the number of auxiliary samples from 1000 to 5000, and compare the performance in different settings. In our experiments, we find that increasing the size of auxiliary sample set would improve the performance of learning tasks, but the improvements are marginal when the size is over 3000. Due to the space limit, we only report the results of self-taught learning algorithms under two settings that use 1000 and 3000 auxiliary images, respectively.

1http://labelme.csail.mit.edu/Release3.0/
Figure 6.3: Convergence property of our approach on Caltech-101 dataset, measured by (a) relative error of $Z_T$; (b) relative error of $D$.

**Target Domain Datasets.** To extensively testify our approach and related methods, we utilize the following five benchmark datasets.

- **MSRC-v1** dataset\(^2\) contains 240 images of 9 classes. Following [217], we choose 7 classes including airplane, bicycle, building, car, cow, face, tree, and each class has 30 images. This dataset owns obvious clutter and variability in the appearances of objects. The **MSRC-v2** dataset is an extension of MSRC-v1. It contains 591 images of 23 object classes. Fig. 6.2 (c) shows some images in the MSRC-v1 dataset.

- **Caltech-101** dataset\(^3\) contains 9411 images of objects, belonging to 102 categories (including one background class). Following [218], we use the 20-class subset includes Faces, Leopards, Motorbikes, Binocular, Brain, Camera, Car-Side, Dollar-Bill, Ferry, Garfield, Hedgehog, Pagoda, Rhino, Snoopy, Stapler, Stop-Sign, Water-Lilly, Windsor-Chair, Wrench, Yin-Yang, and therefore has 1230 images in total. Fig. 6.2 (d) shows some images in the Caltech-101 dataset.

- **Caltech-UCSD Birds** dataset\(^4\) contains the photos of 200 birds species (mostly North American). There are 6033 images in total. In the experiments, we select the first 30 categories. Fig. 6.2 (e) shows some images in the Caltech-UCSD Birds dataset.

---

\(^2\)http://research.microsoft.com/en-us/projects/objectclassrecognition/

\(^3\)http://www.vision.caltech.edu/Image_Datasets/Caltech101/

\(^4\)http://www.vision.caltech.edu/visipedia/CUB-200.html
Scene-15 dataset\(^5\) contains 4485 images spread over 15 natural scene categories. The fifteen scene categories contain 200 to 400 images each and range from natural scenes like mountains and forests to man-made environments like kitchens and offices. Fig. 6.2 (b) shows some images in the Scene-15 dataset.

Table 6.2 summarizes the details of each target dataset. For each dataset, we extract the local binary pattern (LBP) features from each image using the VLFeat package \(^6\), and finally quantize every image as a 928 dimensional feature vector. LBP was selected due to its simple implementation and good performance on image representation. Other types of features could also be used in the proposed framework.

**Baselines.** We compare our S-Low clustering algorithm with several representative subspace clustering methods including Scalable Sparse Subspace Clustering (SSSC) \(^{20}\), LRR \(^{13}\), Latent LRR \(^{29}\) and Fixed Rank Representation (FRR) \(^{181}\). Although the RDSTL \(^{59}\) method is not designed for clustering problems, we also testify its performance on subspace clustering to further illustrate the differences between our approach and RDSTL. We utilize an unsupervised version of RDSTL, by replacing the classifier in RDSTL with the graph construction procedure. In detail, the learned dictionary \(D\) is used for generating new codings of each sample, and then a graph is constructed using the codings. As we have two different auxiliary sample sets, we use RDSTL-A and Ours-A to denote the methods employing 1000 images from the auxiliary domain, and RDSTL-B and Ours-B use the auxiliary sample set with 3000 images.

For the image classification task, we compare our S-Low classification algorithm with supervised learning method SVM \(^{146}\), semi-supervised learning method transductive SVM (TSVM) \(^{220}\), low-rank learning methods Latent LRR \(^{29}\) and FRR \(^{181}\). We also compare our approach with the state-of-the-art domain adaptation and self-taught learning methods, including the Landmarks Selection-based Subspace Alignment (LSSA) method \(^{221}\), STL \(^{55}\), and RDSTL \(^{59}\).

**Evaluation Metrics.** For the clustering task, the whole target data set is taken as input of the algorithm. To evaluate the clustering results, we adopt two widely used performance measure, clustering Accuracy (Acc) and normalized mutual information (NMI), which are defined as follows:

\[
\text{Acc} = \frac{\sum_{i=1}^{n} \delta(\text{map}(r_i), l_i)}{n},
\]

where \(r_i\) denotes the cluster label of sample \(x_i\), \(l_i\) denotes the ground truth, \(n\) is the total number of samples, \(\delta(x, y)\) equals to 1 if and only if \(x = y\), and \(\text{map()}\) is the permutation mapping function.

\(^5\)http://www-cvr.ai.uiuc.edu/ponce_grp/data/
\(^6\)http://www.vlfeat.org/
that maps each cluster label \( r_i \) to the equivalent label from the data set.

\[
\text{NMI}(A, B) = \frac{\text{MI}(A, B)}{\max(H(A), H(B))},
\]

where \( A \) and \( B \) are the predicted clustering index and the ground truth, respectively. \( \text{MI}(A, B) \) denotes the mutual information between \( A \) and \( B \). \( H(A) \) and \( H(B) \) denote the entropies of \( p(a) \) and \( p(b) \).

For the classification task, we follow the standard ways to produce training and test splits on different datasets. Following [59], we conduct 5-fold experiments on the MSRC-v1 dataset. We randomly select 5, 10, 15, and 30 samples to construct training set on the Caltech-101 dataset, and the rest samples for testing. On the Scene-15 dataset, following [222], we randomly select 100 training samples, and the rest samples are used to construct test sample set. We will compute the classification accuracy, and show the confusion matrices.

### 6.4.2 Property Analysis

**Convergence Analysis.** Fig. 6.3(a-b) show our approach converges quickly. The relative errors in Fig. 6.3 (a) and (b) are calculated by \( \frac{\| Z_{T(k+1)} - Z_{Tk} \|_F}{\| Z_{Tk} \|_F} \) and \( \frac{\| D^{j+1} - D^j \|_F}{\| D^j \|_F} \).
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

Figure 6.5: Parameter sensitivity of our approach on Caltech-101 dataset: (a) clustering accuracy with different dictionary sizes; (b) NMI with different values of $K$.

respectively. Especially, Fig. 6.3 (b) shows that our dictionary converges within a few number of iterations, which is beneficial to some large-scale applications.

**Sensitivity of Parameters.** In our approach, there are three main parameters, $\lambda_1$, $\lambda_2$ and $\lambda_3$. To choose proper values for them, we evaluate the parameter sensitivity on the MSRC-v1 dataset. We conducted an internal 5-fold cross-validation in the training set to fine tune the parameters. Fig. 6.4 shows the accuracy of Ours-B approach under different settings of $\lambda_1$, $\lambda_2$ and $\lambda_3$, respectively. Here, $\lambda_1$ and $\lambda_2$ are used to handle the noise in samples, while $\lambda_3$ controls the structural sparsity in $Z_T$, which allows the source data to select some bases from $D$. Fig. 6.4 also shows that our approach achieves relatively stable performance when $\lambda_3$ is altered from 1 to 2, and $\lambda_2$ does not affect the results significantly in the range $[2, 3]$. For all the databases, we fine tune those parameters to achieve their best performance.

The size of dictionary $D$ is another parameter that should be predefined. Fig. 6.5(a) shows the accuracy of different algorithms on the Caltech-101 dataset. It shows that our approach is not sensitive to the size of dictionary. We empirically set the size of dictionary as 200, 200, 200, 300, and 400 on the MSRC-v1, MSRC-v2, Caltech-101, Scene-15 and Caltech-UCSD Birds datasets, respectively. In Algorithm 2, we prune the graph by keeping only $K$ nearest neighbors for each node. Fig. 6.5(b) shows the NMI with different values of $K$. In the following experiments, $K$ is set to 20.

Note that the edge-pruning procedure is not the main reason of the performance improve-
CHAPTER 6. ROBUST DICTIONARY LEARNING FOR KNOWLEDGE TRANSFER

Table 6.3: Subspace clustering accuracies (%) of all compared methods. The version A of RDSTL and our method uses 1000 auxiliary images, and version B uses 3000 images.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSRC-v1</th>
<th>MSRC-v2</th>
<th>Caltech-101</th>
<th>Scene-15</th>
<th>Birds</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRR [13]</td>
<td>70.95</td>
<td>32.08</td>
<td>41.22</td>
<td>36.87</td>
<td>13.46</td>
</tr>
<tr>
<td>SSSC [20]</td>
<td>69.25</td>
<td>33.25</td>
<td>40.01</td>
<td>28.81</td>
<td>18.25</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>71.91</td>
<td>31.37</td>
<td>44.39</td>
<td>32.40</td>
<td>15.20</td>
</tr>
<tr>
<td>FRR [181]</td>
<td>70.48</td>
<td>32.75</td>
<td>42.67</td>
<td>31.32</td>
<td>17.91</td>
</tr>
<tr>
<td>RDSTL-A [59]</td>
<td>52.68</td>
<td>27.16</td>
<td>35.44</td>
<td>27.06</td>
<td>11.05</td>
</tr>
<tr>
<td>Ours-A</td>
<td>74.25</td>
<td>38.42</td>
<td>50.25</td>
<td>43.17</td>
<td>21.63</td>
</tr>
<tr>
<td>Ours-B</td>
<td><strong>75.16</strong></td>
<td><strong>43.21</strong></td>
<td><strong>55.47</strong></td>
<td><strong>46.75</strong></td>
<td><strong>23.91</strong></td>
</tr>
</tbody>
</table>

Table 6.4: NMI of all compared methods. The version A of RDSTL and our method uses 1000 auxiliary images, and version B uses 3000 images.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSRC-v1</th>
<th>MSRC-v2</th>
<th>Caltech-101</th>
<th>Scene-15</th>
<th>Birds</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRR [13]</td>
<td>0.6021</td>
<td>0.3892</td>
<td>0.4697</td>
<td>0.3185</td>
<td>0.2305</td>
</tr>
<tr>
<td>SSSC [20]</td>
<td>0.6128</td>
<td>0.3921</td>
<td>0.4832</td>
<td>0.3305</td>
<td>0.2651</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>0.5939</td>
<td>0.3719</td>
<td>0.4728</td>
<td>0.2932</td>
<td>0.2454</td>
</tr>
<tr>
<td>FRR [181]</td>
<td>0.5932</td>
<td>0.4033</td>
<td>0.4489</td>
<td>0.3271</td>
<td>0.2392</td>
</tr>
<tr>
<td>RDSTL-A [59]</td>
<td>0.3604</td>
<td>0.2915</td>
<td>0.3109</td>
<td>0.2515</td>
<td>0.2101</td>
</tr>
<tr>
<td>RDSTL-B [59]</td>
<td>0.3782</td>
<td>0.2618</td>
<td>0.3675</td>
<td>0.2613</td>
<td>0.2075</td>
</tr>
<tr>
<td>Ours-A</td>
<td>0.6725</td>
<td>0.4778</td>
<td><strong>0.5267</strong></td>
<td>0.3795</td>
<td>0.2811</td>
</tr>
<tr>
<td>Ours-B</td>
<td><strong>0.6841</strong></td>
<td><strong>0.5132</strong></td>
<td>0.5215</td>
<td><strong>0.4015</strong></td>
<td><strong>0.3091</strong></td>
</tr>
</tbody>
</table>

ment. To verify this, we conduct experiments to evaluate the performance of Latent LRR with edge pruning. On the Caltech-101 dataset, the clustering accuracy of Latent LRR is 44.39%. The accuracies of Latent LRR with edge pruning are: 44.57% ($K=3$), 44.86% ($K=5$), 44.12 ($K=10$), where $K$ is the number of neighbors per sample. It shows that edge pruning can slightly enhance the performance, but it is not the main reason.

6.4.3 Clustering Results

Table 6.3 shows the subspace clustering accuracies of all compared methods on the five datasets, and Table 6.4 shows the corresponding NMI for each method. We can observe that all the low-rank based methods outperform RDSTL, since they explicitly consider the structure information in sample space. When class labels are not available, the underlying structure information of data plays an important role in learning tasks. Latent LRR, which models the effect of hidden data, performs better than LRR and FRR. By virtue of a more informative dictionary learnt from both
auxiliary domain and target domain, our approach achieves better performance than other competitors on all the five datasets.

In addition, by increasing the data size in auxiliary domain, the performance of self-taught learning methods could be slightly improved, as RDSTL-B and Ours-B outperform RDSTL-A and Ours-A in most cases, respectively. We also noticed that, on the Caltech-UCSD Birds dataset, unsupervised clustering is a rather challenging problem, and the clustering accuracies of all compared methods are a bit low. One possible reason is that most categories share many common visual elements such as different birds in the wild.

### 6.4.4 Classification Results

Our S-Low classification algorithm is compared with SVM [146], transductive SVM (TSVM) [220], Latent LRR [29], FRR [181], LSSA [221], STL [55], and RDSTL [59]. We also have two versions for STL, RDSTL, and our method that use two auxiliary sample sets, respectively.

STL, LSSA and our approach need to train SVM classifiers. For SVM and TSVM, we utilize the Gaussian kernel (i.e., \( K(x_i, x_j) = \exp(-\beta \|x_i - x_j\|^2) \)), and tune the parameter \( \beta \) and regularization parameter \( C \) in the range of \( \{10^{-5}, \cdots, 10^{-1}, 1, 10^1, \cdots, 10^5\} \) to obtain their best classification results. For each method, an internal 5-fold cross-validation in the training set is conducted to fine tune the parameters.
Table 6.5: Average classification accuracies (%) of all compared methods on three datasets.

(a) MSRC-v1 and Scene-15 datasets

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSRC-v1</th>
<th>Scene-15</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM [146]</td>
<td>79.62</td>
<td>76.41</td>
</tr>
<tr>
<td>TSVM [220]</td>
<td>79.84</td>
<td>75.35</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>81.90</td>
<td>62.53</td>
</tr>
<tr>
<td>FRR [181]</td>
<td>80.45</td>
<td>60.65</td>
</tr>
<tr>
<td>LSSA [221]</td>
<td>81.59</td>
<td>72.61</td>
</tr>
<tr>
<td>STL-A [55]</td>
<td>83.04</td>
<td>73.70</td>
</tr>
<tr>
<td>STL-B [55]</td>
<td>83.62</td>
<td>75.12</td>
</tr>
<tr>
<td>RDSTL-A [59]</td>
<td>89.11</td>
<td>77.08</td>
</tr>
<tr>
<td>RDSTL-B [59]</td>
<td>89.44</td>
<td>78.52</td>
</tr>
<tr>
<td>Ours-A</td>
<td>91.52</td>
<td>82.45</td>
</tr>
<tr>
<td>Ours-B</td>
<td><strong>92.36</strong></td>
<td><strong>82.73</strong></td>
</tr>
</tbody>
</table>

(b) Caltech-101 dataset

<table>
<thead>
<tr>
<th>Methods</th>
<th>5 train</th>
<th>10 train</th>
<th>15 train</th>
<th>30 train</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM [146]</td>
<td>45.53</td>
<td>53.61</td>
<td>57.72</td>
<td>67.08</td>
</tr>
<tr>
<td>TSVM [220]</td>
<td>44.18</td>
<td>52.78</td>
<td>57.35</td>
<td>65.83</td>
</tr>
<tr>
<td>LatLRR [29]</td>
<td>46.32</td>
<td>53.29</td>
<td>58.15</td>
<td>68.67</td>
</tr>
<tr>
<td>FRR [181]</td>
<td>45.21</td>
<td>53.57</td>
<td>58.63</td>
<td>67.52</td>
</tr>
<tr>
<td>LSSA [221]</td>
<td>45.10</td>
<td>54.92</td>
<td>58.25</td>
<td>70.33</td>
</tr>
<tr>
<td>STL-A [55]</td>
<td>47.60</td>
<td>54.73</td>
<td>59.06</td>
<td>71.46</td>
</tr>
<tr>
<td>STL-B [55]</td>
<td>47.92</td>
<td>55.07</td>
<td>59.54</td>
<td>71.31</td>
</tr>
<tr>
<td>RDSTL-A [59]</td>
<td>49.54</td>
<td>56.84</td>
<td>61.26</td>
<td>72.62</td>
</tr>
<tr>
<td>RDSTL-B [59]</td>
<td>50.13</td>
<td>57.05</td>
<td>61.73</td>
<td>72.95</td>
</tr>
<tr>
<td>Ours-A</td>
<td>53.28</td>
<td><strong>58.92</strong></td>
<td>63.95</td>
<td>74.51</td>
</tr>
<tr>
<td>Ours-B</td>
<td><strong>53.16</strong></td>
<td><strong>59.33</strong></td>
<td><strong>65.12</strong></td>
<td><strong>74.78</strong></td>
</tr>
</tbody>
</table>

The dictionary $D$ plays a key role in our approach, and one interesting question is that, will the size of dictionary greatly influence the classification accuracy? Fig. 6.6 shows the classification accuracy of Ours-A approach when the size of dictionary varies from 100 to 800 on MSRC-v1 and Scene-15 datasets, respectively. It shows that our approach obtains relatively similar results on the Scene-15 dataset with different dictionary sizes, and obtains better performance on the MSRC-v1 dataset when the dictionary size is 200. In our experiments, we empirically set the size of dictionary as 200 on the MSRC-v1 and Caltech-101 datasets, and 400 on the Scene-15 dataset.

All compared methods are repeated 10 times on the Caltech-101 and Scene-15 datasets. Table 6.5(a) shows the average classification results on the MSRC-v1 dataset, and Table 6.5(b) shows the results on the Caltech-101 dataset under different settings. To take a close look at our success and
Figure 6.7: (a-b) Confusion matrices of our approach on MSRC-v1 and Scene-15 datasets. (c) Example images from classes with high classification accuracy of Caltech-101.

failure cases, Fig. 6.7(b) and (c) show the confusion matrices of our approach on the MSRC-v1 and Scene-15 datasets, respectively, and Fig. 6.7(a) provides examples from classes with high accuracy on the Caltech-101 dataset, when the number of training samples is 30.

We can observe from Table 6.5 that the advantages of self-taught learning are extensively demonstrated, since all the three self-taught learning methods (i.e., STL, RDSTL and our approach) outperform other competitors. The domain adaption methods like LSSA usually assume that the auxiliary and target domains share similar learning tasks. However, such an assumption does not hold in the setting of self-taught learning, i.e., the auxiliary domain may contain arbitrary images. Thus, we observe that LSSA cannot outperform the self-taught learning methods. Furthermore, our
approach consistently performs better than STL and RDSTL, and the main reason is that our low-rank codings could provide robust representations for images by using an informative dictionary.

6.4.5 Discussions

To illustrate how low-rank constraints and auxiliary data help the learning tasks in the target domain, in Fig. 6.8, we visualize several graphs learned by Latent LRR, RDSTL and Ours-B approach on the MSRC-v1 image dataset, respectively. We have the following observations.

• Due to the low-rank constraint, the graphs learned by Latent LRR and our approach shown in Fig. 6.8 (a) and (c) have a block-diagonal structure. However, the RDSTL graph shown in Fig. 6.8 (c) does not have such a structure, as the global structure information is not considered in the model.

• Compared with the Latent LRR graph, our graph is sparser, and has a clearer block-diagonal structure. The reasons are two-fold. First, Latent LRR uses the sample set itself as a dictionary, while our approach learns an informative dictionary for representation. Second, our approach prunes the edges with small weights to produce a sparse graph, while the graph learned by Latent LRR is very dense.

• The weights in our graph (illustrated as red color in Fig. 6.8 (c)) are much higher than those in Latent LRR (illustrated as blue color in Fig. 6.8 (a)). With the help of auxiliary dataset, the results in Table 6.3 and Table 6.4 validate the effectiveness of our graph.
6.5 Summary

In this chapter, we propose a novel self-taught low-rank coding approach for clustering visual data. Our approach jointly learns a dictionary by virtue of rich information from auxiliary domain, and robust low-rank representations for target domain. We derive both unsupervised and supervised learning algorithms for subspace clustering and image classification. Experimental results on five benchmark datasets demonstrate the effectiveness of our approach compared with the state-of-the-art self-taught learning methods.

There remain several interesting directions for our future work: (1) we would design S-Low coding based classification approaches, (2) given a training set in target domain, we may automatically choose samples from the auxiliary domain, (3) we would provide fast solutions to our framework by using the divide-and-conquer technique.
Chapter 7

Conclusion

Representation learning plays an essential role in extracting knowledge from visual data. Practical visual learning algorithms are expected to effectively and efficiently process high-dimensional visual data such as real-world images and videos. Numerous applications can be benefited from the effective feature representations, such as self-driving cars, video surveillance systems, and virtual reality systems.

This dissertation studies three types of data representations, including graph, subspace and dictionary, and presents a set of robust representation learning approaches. By leveraging the low-dimensional subspace structures in the high-dimensional feature space, the proposed approaches are able to capture the intrinsic characteristics from noisy observations, and to extract very compact feature representations for specific learning tasks. In Chapter 2, two graph construction approaches are designed to build unbalanced and balanced graphs, respectively, in the low-rank coding space. We reveal that the similarity measurement between every two samples in the low-rank coding space is more robust than that in the sample space. Remarkable performance are observed when applying our graphs to image clustering and semi-supervised classification. In Chapter 3, a robust subspace discovery approach is designed by integrating the merits of subspace learning and low-rank matrix recovery. Our approach learns a robust and discriminative subspace for dimensionality reduction and classification. Experimental results show that, when the images are heavily corrupted, our approach can still achieve promising classification accuracy. In Chapter 4, a robust multi-view subspace learning approach is presented for time series classification. In order to preserve the temporal structure of time series data, our approach adopts bilinear projections that map high-dimensional time series data into a low-dimensional subspace. Moreover, a pair of bilinear projections are learned for each data view, and the data from multiple views are mapped into a shared latent space. A
CHAPTER 7. CONCLUSION

discriminative regularization and a temporal smoothness constraint are also incorporated during the mapping. Empirically evaluations on real-world multi-view time series datasets validate the effectiveness of our approach. In Chapter 5, a non-negative dictionary learning approach is designed for human motion segmentation. A least-square regression based formulation is adopted to learn dictionary and also the compact codings for the video sequence. To obtain more expressive codings, we learn a non-negative dictionary from data, instead of using the data self-expressive models. A temporal Laplacian regularization function to encode the sequential relationships in time series data. Experimental results on several human action and gesture datasets show that our approach outperforms the traditional temporal clustering baselines. In Chapter 6, a robust dictionary learning approach is proposed for transferring knowledge from an auxiliary domain to a target domain. The target domain contains a few training images, while the auxiliary domain has a large amount of irrelevant images. Our approach jointly learns a dictionary from two domains, and utilizes a constraint to enhance the reconstruction fidelity in the target domain. Extensive experiments on five image datasets demonstrate the effectiveness of our approach.

In addition to visual learning, we have applied robust representation learning models discussed in this dissertation to other applications [223], such as outlier detection [37], collaborative filtering [224], response prediction [225], and causal inference [226]. In the future work, we would like to explore the possibility of applying our robust representation learning models to other application domains, such as medical image analysis, bioinformatics, and user behavior modeling. Moreover, we will design intelligent learning systems which could extract high-level knowledge from visual scenes, such as question answering and causal inference.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


BIBLIOGRAPHY


[225] ——, “Predicting user behavior in display advertising via dynamic collective matrix factor-
ization,” in Proceedings of the 38th International ACM SIGIR Conference on Research and

of treatment effects in digital marketing campaigns,” in Proceedings of the Twenty-Fifth