FEATURE SELECTION AND DATA RECONSTRUCTION VIA ROBUST AND FLEXIBLE LEARNING MODELS

by

DI MING

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2020

To my parents, wife, and daughter.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervising professor Dr. Chris Ding for constantly teaching, motivating, encouraging me, and also for his invaluable advices. Without his helps and instructions, I can not finish my degree and thesis. I also would like to express my gratitude to Dr. Jean Gao, Dr. Junzhou Huang, and Dr. Dajiang Zhu for their constructive suggestions on my research, and taking time to serve in my dissertation committee.

I also would like to extend my appreciation to my colleagues in the Data Science Lab at the Department of Computer Science and Engineering. It is my great pleasure to have this opportunity to work with so many smart people in my group, and I have learned a lot from technical discussions with them. I also thank my friends who have helped me a lot throughout the career.

Finally, I would like to express sincere gratitude to my parents who have consistently encouraged and inspired me during my studies. Without their love, sacrifice, patience, and support, it would not be possible for me to reach this stage in my career. I am also deeply grateful to my wife for her patience and sacrifice. No matter what kind of challenges, difficulties, and painful situations I was confronted with, she is always on my side and supports me.

January 24, 2020

ABSTRACT

FEATURE SELECTION AND DATA RECONSTRUCTION VIA ROBUST AND FLEXIBLE LEARNING MODELS

Di Ming, Ph.D.

The University of Texas at Arlington, 2020

Supervising Professor: Chris Ding

Feature selection and data reconstruction are very important topics in machine learning area. In today's big data environment, many data could have high dimensions and come with noise, corruption, etc. Thus, we develop robust and flexible learning models so as to select the relevant features from the high-dimensional data spaces and reconstruct the original clean data from the corrupted input data more efficiently and more effectively.

To resolve the inflexibility of the widely used class-shared feature selection methods such as $\ell_{2,1}$-norm, we derive LASSO from probabilistic selection on ridge regression which provides an independent point of view from the usual sparse coding point of view, and further propose the probability-derived $\ell_{1,2}$-norm based feature selection to select discriminative features. On the other hand, we propose a novel "exclusive $\ell_{2,1}$" regularization to select robust and flexible feature. Exclusive $\ell_{2,1}$ regularization brings out joint sparsity at inter-group level and exclusive sparsity at intra-group level simultaneously. As a result, it combines the advantages of both $\ell_{2,1}$-norm (increase the robustness) and $\ell_{1,2}$-norm (provide the flexibility) regularizations together.

For purpose of automatically recovering the original clean data from the noisy input in unsupervised fashion, we propose a deep robust data reconstruction method in the form of autoencoder networks using ℓ_1 loss, and introduce a smoothed ReLU (sReLU) activation function to resolve the black spot problem in the outputs of the network naively using ℓ_1 loss with popular ReLU. In addition, we propose a robust PCA based low-rank and sparse data reconstruction method, and theoretically prove the underlying connection between the regularization and the robustness. Towards resolving the corresponding multivariate optimization problem efficiently, we introduce an "exact solver" based optimization algorithm to minimize robust L_1-PCA models via alternative optimization strategy.

Experimental result on benchmark datasets shows: (i) the feature selected by robust and flexible learning models achieves a higher accuracy in classifying the multi-class data; (ii) the data reconstructed by robust and flexible learning models obtains a smaller noise-free error in recovering the corrupted noise data. Thus it can be seen that the proposed robust and flexible learning models obtain better performance than state-of-the-arts in real-world applications.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Today is an era of information explosion. The amount of data is increasing rapidly. Thus, data analysis becomes very important to many real-world applications, since it can discover useful information from the massive data to support the decision-making. The whole process of data analysis includes inspecting, cleansing, transforming, and modeling data. In the thesis, we develop novel robust and flexible models for feature selection and data reconstruction, which focuses on improving the first two steps of data analysis. As a result, our works can help users to build better machine learning models, and further improve the performance on many real-world applications, such as classification, clustering, etc.

In today's big data environment, many data has high-dimensions, for example bio-microarray datasets with more than 10,000 features/genes [1] are commonplace. Thus, selecting a subset of useful features is a very important topic in machine learning area. Feature selection, also known as variable selection, attribute selection, or variable subset selection, is the process of identifying relevant/significant features and screening out irrelevant/redundant noise features. The goal of feature selection is as follows: (i) reduces the data sizes, so as to shorten training and testing times; (ii) avoid the curse of dimensionality; (iii) enhance generalization by reducing overfitting; (iv) simplifies the interpretation of the results, for example, gene-expression analysis [2], proteomic biomarkers discovery [3], molecular cancer prediction [4]; (v) improve the performance on real-world applications such as classification.

Many real life data come with noise, corruption, etc. For example, photos taken through a window are often compromised by dirt or rain on the window surface [5]; As a result, data reconstruction/recovery is the very important topic in machine learning and computer vision areas. The task aims at automatically recovering the original clean data from the corrupted input in an unsupervised manner, also known as denoising [6] and noise reduction [7] in other domains. The goal of data reconstruction is as follows: (i) removes noise so as to improve the quality of the data; (ii) captures the complicated intrinsic property of the noisy data; (iii) learns a lower-dimensional latent representation of the data, which leads to easier learning and easier visualization with fewer parameters; (iv) improve the performance on real-world applications such as clustering.

As it can be seen the first two key steps of data analysis (i.e. data inspecting and data cleaning) are of significant importance to users who want to build a better model given the raw data. However, existing methods have some limitations and problems, especially when data have high dimensions and noises. Thus, we are interested in developing robust and flexible models to improve the efficiency and the effectiveness of feature selection and data reconstruction. To resolve the inflexibility of class-shared feature selection approaches, we derive LASSO from a probabilistic point of view, and use the probability-derived $\ell_{1,2}$-norm to select discriminative features, which can provide certain flexility for each class. In addition, "exclusive $\ell_{2,1}$" regularization is proposed to conduct robust flexible feature selection, which can increase the robustness for using $\ell_{1,2}$-norm alone and provide the flexibility for using $\ell_{2,1}$-norm along. On the other hand, linear models based on nuclear norm and other low-rank methods are good at removing simple synthetic occlusions such as black rectangles, but cannot achieve a good performance on complicated occlusions, for example sunglasses and scarf in AR faces [8]. Thus, we propose a deep robust data reconstruction method using ℓ_1-autoencoder networks to capture complicated intrinsic property of the corrupted input data. Besides, we present a robust PCA based low-rank and sparse data reconstruction method, and propose an "exact solver" based optimization algorithm to further improve the robustness of models and the quality of reconstructions.

1.2 Contribution

1.2.1 Probabilistic LASSO

We first propose selective ridge regression to measure the importance of each feature, and then derive LASSO from a probabilistic point of view. Based on probability-derived $\ell_{1,2}$-norm, we introduce a ranking method to select feature using a probabilistic selection vector. Furthermore, we point out an interesting point that selecting less number of features in selective ridge regression leads to the increase of regularization strength in LASSO, which explains the reason why the weight variable has more zero values with a larger hyperparameter in front of ℓ_1-norm. However, LASSO is a regression analysis method for two-class problem. Towards extending probabilistic LASSO to multi-class problems, we add a probabilistic selection vector for each class separately. Thus, we can apply $\ell_{1,2}$-norm regularization to select discriminative features for each class and provide certain flexility. That is to say, selected features are not vigorously exact same for different classes. Experimental results on six benchmark datasets (3 image and 3 bio-microarray datasets) show that our flexible feature selection method via probability-derived $\ell_{1,2}$-norm has a better performance on multi-class classification as compared to state-of-the-art algorithms.

1.2.2 Exclusive $\ell_{2,1}$ Regularization

To combine the advantages of different sparsity-induced norms such as $\ell_{2,1}$-norm and $\ell_{1,2}$-norm, a novel "exclusive $\ell_{2,1}$" regularization (short for $\ell_{2,1}$ with exclusive LASSO) is proposed to conduct robust flexible feature selection. Exclusive $\ell_{2,1}$ regularization brings out joint sparsity at inter-group level and exclusive sparsity at intra-group level simultaneously. Thus, it not only removes irrelevant noise features (i.e. increase the robustness via $\ell_{2,1}$-norm), but also selects discriminative features for each class (i.e. provide the flexibility via $\ell_{1,2}$-norm). Towards a better understanding of exclusive sparsity, an efficient sorting-based explicit approach is proposed to solve $\ell_{1,2}$-norm based proximal operator-type problem. We further point out an interesting property of $\|\mathbf{w}\|_1^2$ regularization as compared to standard $\|\mathbf{w}\|_1$ regularization. As the regularization strength (i.e. the hyperparameter in front of regularization term) increases, $\|\mathbf{w}\|_1$ regularization shrinks all the elements in $\mathbf{w}$ to zero, while $\|\mathbf{w}\|_1^2$ regularization will leave at least one element in $\mathbf{w}$ as nonzero. This is so-called exclusive sparsity of $\ell_{1,2}$-norm: all the elements in a vector are competing with each other, and finally only one element can survive with nonzero value. Then, an efficient augmented Lagrangian multiplier based optimization algorithm is proposed to solve the exclusive $\ell_{2,1}$ regularization in a row-wise fashion, which greatly reduces the computational cost and is well-suited for large-scale data. Experimental results on twelve benchmark datasets (4 image, 1 spoken letter recognition, 5 bio-microarray, and 2 text datasets) validates the effectiveness of our robust flexible feature selection method on multi-class classification as compared to state-of-the-arts.

1.2.3 Deep ℓ_1-Autoencoder Networks

To remove the noise/corruption/occlusion from the input data in real-world applications, we propose a deep autoencoder network with ℓ_1 loss to conduct robust data reconstruction. Compared to state-of-the-art linear models and low-rank methods, our proposed multi-layer network is more capable of capturing the complicated intrinsic property of the corrupted input data. An interesting finding in this work is that:

(i) autoencoder using ReLU with L_1 loss produces output with black spots; (ii) autoencoder using ReLU with cross-entropy loss or L_2 loss produce outputs (which is the desired reconstructed images) without black spots. Then, the analysis of the gradient of loss and activation functions motivates us to introduce a smoothed ReLU (sReLU) activation for resolving the black spot problem associated with ℓ_1+ReLU networks. Extensive experiments on two benchmark datasets are performed to verify the effectiveness of our deep robust reconstruction method against various kinds of occlusions such as circle, rectangle, cross, and sunglasses, which outperforms state-of-the-arts with lower noise-free reconstruction errors. Additionally, experimental results show that increasing number of layers in ℓ_1-autoencoder networks with smoothed ReLU as activation steadily improves the quality of reconstructed images.

1.2.4 Robust L_1-PCA

For purpose of automatically removing the noise in the data, we propose a robust PCA based low-rank and sparse data reconstruction method in an unsupervised fashion. Traditional robust PCA models usually have an assumption of the underlying noises in the original feature space. More than that, our proposed robust reconstruction method also models the underlying noises of principal directions and components in the low-dimensional latent feature space. We further derive two tight upper bounds of robust L_1-PCA models, which theoretically proves the connection between robustness and regularization. For solving L_1-/L_{21}-norm penalty based robust L_1-PCA models, we introduce two alternative optimization algorithms. Firstly, an augmented Lagrangian multiplier based optimization algorithm is presented to decompose original problem into several subproblems, which can be resolved in a matrix-based fashion with close-form solutions. To further improve the robustness of reconstruction models and resolve the early-stopping problem of ALM based optimization algorithms, we introduce an "exact solver" based optimization algorithm, which minimizes the objective function with respect to a single entry of principal directions or components at each time. Most importantly, exact solver can obtain the globally optimal solution for each entry while fixing others in linear time. Experimental results on benchmark dataset show the proposed robust L_1-PCA model achieves better results on reconstructing the corrupted images as compared to state-of-the-arts, and exact solver based optimization algorithm recovers the original clean image with better quality and further improves the robustness of reconstruction models as compared to the widely used ALM based optimization algorithm.

1.3 Organization

The rest of the thesis is organized as follows. At first, a probabilistic derivation of LASSO and $\ell_{1,2}$-norm feature selection is introduced in Chapter 2. Secondly, robust flexible feature selection via exclusive $\ell_{2,1}$ regularization is introduced in Chapter 3. Next, deep robust data reconstruction using ℓ_1-autoencoder networks is introduced in Chapter 4. In the following, robust PCA based low-rank and sparse data reconstruction is introduced in Chapter 5. Finally, conclusion are given in Chapter 6.

CHAPTER 2

A PROBABILISTIC DERIVATION OF LASSO AND L12-NORM FEATURE SELECTIONS

2.1 Introduction

Feature selection is one of important tasks of machine learning. In today's big data environment, many data has high-dimensions, e.g., biology datasets with around 10k features/genes [1] are commonplace. Selecting useful set of features reduces the data sizes, improves many learning models such as classification, regression, etc, and more importantly simplifies the interpretation of machine learning results, with many applications in gene-expression analysis [2], proteomic biomarkers discovery [3], molecular cancer prediction [4].

Feature selection has been widely investigated in many applications with great assistance to practical performance. The main focus in the literature is on the supervised learning, which evaluates the relevance between features and class labels. The evaluation metric divides feature selection algorithms into three main categories [9], which are filter, wrapper, and embedded methods. Independent of any specific learning models, filter-type methods such as F-statistic [10] and ReliefF [11] can quickly select features which are most correlated with class labels. However, redundant features are usually remained in the subset of selected features via aforementioned algorithms. Thus, mRMR [12] is proposed to maximize relevance and minimize redundancy simultaneously, which can effectively overcome the shortage of previous methods and further improve the practical performance. On the contrary, wrapper-type methods such as SVM-RFE [13] are dependent on a specific classifier to iteratively search the best feature subset, but which has extremely high computational cost and potential overfitting risk.

Recently, sparse coding based methods (also called embedded methods) become popular in the study of feature selection. This approach combines the advantages of above-mentioned two kinds of methods. The sparse model tries to find a compromise between loss and penalty, e.g., classic Lasso [14] using ℓ_1-norm constraint, which is also known as sparse coding in dictionary learning. To remove redundant noise features, ℓ_1-SVM [15] is introduced to generate the sparse solution for two-class classification problem. On the other hand, in multi-task setting, researchers [16, 17, 18, 19] focus on designing a collaborative model to select class-shared features via $\ell_{2,1}$-norm, which is first proposed in [20] as rotational invariant ℓ_1-norm for purpose of robust subspace factorization. During the same period, $\ell_{1,\infty}$-norm [21] is proposed to build a set of jointly sparse models, by means of ℓ_1-ball projection [22]. Besides, sparse coding based method has been applied to many other domains, such as sparse subspace learning [23], sparse representation based classification [24], etc.

2.1.1 A Probabilistic View of LASSO

The ℓ_1 based LASSO and the closely related $\ell_{1,2}$-norm feature selection are, in some sense, a prescription using sparse coding. In this work, we show they can be derived from a probability framework, thus provides a strong probabilistic foundation.

We further propose to use this probability-derived $\ell_{1,2}$-norm feature selection. In this approach, features selected from different classes are not vigorously enforced to be exactly same.

However, most of feature selection methods, such as the widely used $\ell_{1,2}$-norm, aim at searching features across all the data instances with joint sparsity, which then enforces the selected features to be exactly same for all classes.

Here, we argue that it is better to allowing selected features to have certain flexibility, not exactly same. In applications and real data, different classes could have its own characteristics, e.g., cars and cups have different features. Thus, using vigorously same set of features is not a natural way to pre-process/prescreen the data. Motivated by exclusive feature learning [25, 26, 27, 28], in this work we propose a flexible feature selection method via $\ell_{1,2}$-norm regularization. In previous works, $\ell_{1,2}$-norm is used to either capture the negative correlation which creates competitions between features across all the classes, or eliminate strongly correlated features in two-class setting. Thus it can be seen that our proposed method has a clear difference from them, since we enforce $\ell_{1,2}$-norm on features for each class to select a subset of important features which are most correlated with each class separately. Using the flexible $\ell_{1,2}$-norm feature selection obtains features that generally perform better in many real datasets, including images and bio-microarray data.

The main contributions of this work include: (1) a probabilistic derivation of LASSO and $\ell_{1,2}$-norm, and illustrating how $\ell_{1,2}$-norm is used to measure the importance of features for each class; (2) an effective algorithm with rigourous convergence analysis is proposed to compute/select the features using $\ell_{1,2}$-norm regularization, which is a parameter-free method and quickly converges; (3) experimental results on six benchmark datasets (including images and bio-microarray data) show that our proposed flexible feature selection method has an overwhelmed advantage over state-of-the-arts.

2.2 Notations and Definitions

Lower-case letters refer to scalars, boldface lower-case letters refer to vectors, and boldface capital letters refer to matrices. n refers to the number of data instances. d refers to the number of features or data dimensions. k refers to the number of classes. The i-th element of vector $\mathbf{w}$ is represented by w_i. The i-th row and j-th column of matrix $\mathbf{W} = (W_{i,j})$ are denoted as $\mathbf{w}^i$ and $\mathbf{w}_j$, respectively. Given a matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$, the Frobenius-norm of matrix $\mathbf{W}$ is $\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^k W_{ij}^2}$. In general, the $\ell_{p,q}$ norm of $\mathbf{W}$ is defined as

$$\|\mathbf{W}\|_{p,q} = \left(\sum_{j=1}^{k} \left(\sum_{i=1}^{d} |W_{ij}|^p \right)^{q/p} \right)^{1/q},$$

with the computational mathematics convention that ℓ_p norm on the first (fastest index) i and ℓ_q norm on the second fast index j. With this convention, the $\ell_{2,1}$-norm based feature selection uses $\|\mathbf{W}^T\|_{2,1}$ regularization; the $\ell_{1,2}$-norm based feature selection uses $\|\mathbf{W}\|_{1,2}$ regularization; the exclusive LASSO uses $\|\mathbf{W}^T\|_{1,2}$ regularization.

2.3 A Probabilistic Derivation of LASSO and $\ell_{1,2}$-Norm Feature Selection

First, the variables of the feature selection model are defined as follows. Training data of n labeled feature vectors are denoted as $\mathbf{X} \in \mathbb{R}^{d\times n} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$, where $\mathbf{x}_i \in \mathbb{R}^d$. The corresponding class labels are denoted as $\mathbf{Y} \in \mathbb{R}^{n\times k} = (\mathbf{y}_1, \cdots, \mathbf{y}_k)$, where $\mathbf{y}^i \in \mathbb{R}^k$ represents the class label for $\mathbf{x}_i$ using one-hot encoding vector, i.e., if $\mathbf{x}_i$ belongs to j-th class, $Y_{ij} = 1$; otherwise, $Y_{ij} = 0$. Weights to be learned for the model are denoted as $\mathbf{W} \in \mathbb{R}^{d\times k} = (\mathbf{w}_1, \cdots, \mathbf{w}_k)$, where $\mathbf{w}_i \in \mathbb{R}^d$ represents the coefficients/features correlated with i-th class.

Our starting point is the LASSO type feature selection formalism using $\ell_{1,2}$-norm:

$$\min_{\widehat{\mathbf{W}}} \quad \left\|\mathbf{X}^T\widehat{\mathbf{W}} - \mathbf{Y}\right\|_F^2 + \lambda \left\|\widehat{\mathbf{W}}\right\|_{1,2}^2 \tag{2.1}$$

Here we use $\widehat{\mathbf{W}}$ to distinguish it from the following presentation.

We now present a new derivation of Eq.(2.1) from a probabilistic selection based on ridge regression. We first expand Eq.(2.1) on $\widehat{\mathbf{W}} = (\widehat{\mathbf{w}}_1, \cdots, \widehat{\mathbf{w}}_k)$

$$\min_{\widehat{\mathbf{W}}} \sum_{j=1}^{k} \left(\left\| \mathbf{X}^T \widehat{\mathbf{w}}_j - \mathbf{y}_j \right\|_2^2 + \lambda \|\widehat{\mathbf{w}}_j\|_1^2 \right) \tag{2.2}$$

Now, we introduce a selection probability vector $\boldsymbol{\theta}_j$ for class j and propose a selection formalism

$$\begin{aligned} &\min_{\mathbf{W},\boldsymbol{\Theta}} \sum_{j=1}^{k} \left(\left\| \mathbf{X}^T (\boldsymbol{\theta}_j^{\frac{1}{2}} \odot \mathbf{w}_j) - \mathbf{y}_j \right\|_2^2 + \lambda \|\mathbf{w}_j\|_2^2 \right) \\ &s.t. \quad \boldsymbol{\theta}_j \geq 0, \mathbf{1}^T \boldsymbol{\theta}_j = 1, \qquad j = 1, \cdots, k, \end{aligned} \tag{2.3}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_k) \in \mathbb{R}^{d \times k}$, $\mathbf{1}$ is a vector of all 1's with appropriate size, and $\odot$ is a element-wise Hadamard product, i.e., $(\mathbf{a} \odot \mathbf{b})_i = a_i b_i$.

Both the optimization problems of Eq.(2.2) and Eq.(2.3) are convex and have unique optimal solutions.

Theorem 2.3.1. *Optimization problems Eq.(2.2) and Eq.(2.3) are equivalent. (A) Once the optimal solution $\{\widehat{\mathbf{w}}_j^*\}$ for Eq.(2.2) is obtained, the optimal solution for Eq.(2.3) is given by*

$$\Theta_{ij}^* = \frac{\left| \widehat{W}_{ij}^* \right|}{\|\widehat{\mathbf{w}}_j^*\|_1}, \ \mathbf{w}_j^* = (\boldsymbol{\theta}_j^*)^{-\frac{1}{2}} \odot \widehat{\mathbf{w}}_j^* \tag{2.4}$$

where $i = 1, \cdots, d$ is the feature/dimension index. (B) On the other direction, once $\{\boldsymbol{\theta}_j^*, \mathbf{w}_j^*\}$ for Eq.(2.3) is obtained, the optimal solution for Eq.(2.2) are given by $\widehat{\mathbf{w}}_j^* = (\boldsymbol{\theta}_j^*)^{\frac{1}{2}} \odot \mathbf{w}_j^*$.

The proof of this theorem is given in Lemma 2.4.1.

2.4 LASSO, Nonnegative Garrote and Selective Ridge Regression

Here we discuss LASSO [14], "selective" ridge regression (see Eq.(2.3)) and nonnegative Garrote of Breiman [29].

In optimization problems Eq.(2.2) and Eq.(2.3), different classes are in fact decoupled. Thus we can optimize them one class at a time. Thus the optimization of Eq.(2.2) is, in essence, equivalent to the following form (we ignore the index j)

$$\min_{\widehat{\mathbf{w}}} \quad \left\|\mathbf{X}^T\widehat{\mathbf{w}} - \mathbf{y}\right\|_2^2 + \lambda\|\widehat{\mathbf{w}}\|_1^2, \tag{2.5}$$

This is LASSO, except that the ℓ_1 term is squared which does not affect the sparsity of $\widehat{\mathbf{w}}$.

Optimization problem Eq.(2.3) is in essence what we would call the "selective" ridge regression

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\theta}} \quad & \left\|\mathbf{X}^T(\boldsymbol{\theta}^{\frac{1}{2}} \odot \mathbf{w}) - \mathbf{y}\right\|_2^2 + \lambda\|\mathbf{w}\|_2^2 \\ s.t. \quad & \boldsymbol{\theta} \geq 0, \mathbf{1}^T\boldsymbol{\theta} = 1. \end{aligned} \tag{2.6}$$

This formulation in some sense is close to the nonnegative Garrote:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \left\|\mathbf{X}^T(\boldsymbol{\theta} \odot \mathbf{w}^0) - \mathbf{y}\right\|_2^2 \\ s.t. \quad & \boldsymbol{\theta} \geq 0, \mathbf{1}^T\boldsymbol{\theta} \leq h, \end{aligned} \tag{2.7}$$

where

$$\mathbf{w}^0 = \arg\min_{\mathbf{w}}\|\mathbf{X}^T\mathbf{w} - \mathbf{y}\|_2^2 \tag{2.8}$$

is the solution to ordinary least squares estimation, and $h \leq 1$ is a constant. In both Eq.(2.6) and Eq.(2.7), the selection vector $\boldsymbol{\theta}$ has similar sparsity pattern of the LASSO.

Lemma 2.4.1. *Optimization problems Eq.(2.5) and Eq.(2.6) are equivalent.*

Proof Starting from Eq.(2.6), we introduce a new variable $\widehat{\mathbf{w}} = \boldsymbol{\theta}^{\frac{1}{2}} \odot \mathbf{w}$, then $\mathbf{w} = \boldsymbol{\theta}^{-\frac{1}{2}} \odot \widehat{\mathbf{w}}$. Thus, the optimization problem (2.6) is transformed into

$$\begin{aligned} \min_{\widehat{\mathbf{w}},\boldsymbol{\theta}} \quad & \left\|\mathbf{X}^T\widehat{\mathbf{w}} - \mathbf{y}\right\|_2^2 + \lambda\sum_{i=1}^{d}\left(\frac{\widehat{w}_i^2}{\theta_i}\right) \\ s.t. \quad & \boldsymbol{\theta} \geq 0, \mathbf{1}^T\boldsymbol{\theta} = 1. \end{aligned} \tag{2.9}$$

When $\widehat{\mathbf{w}}$ is fixed, solving problem (2.9) with respect to $\boldsymbol{\theta}$ is

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{i=1}^{d}\left(\frac{\widehat{w}_i^2}{\theta_i}\right) \\ s.t. \quad & \boldsymbol{\theta} \geq 0, \mathbf{1}^T\boldsymbol{\theta} = 1, \end{aligned} \tag{2.10}$$

which can be solved using Lagrangian multiplier [30]. The optimal solution of $\boldsymbol{\theta}$ is computed as

$$\theta_i = \frac{|\widehat{w}_i|}{\sum_{i'=1}^{d} |\widehat{w}_{i'}|} = \frac{|\widehat{w}_i|}{\|\widehat{\mathbf{w}}\|_1}, \tag{2.11}$$

where $i = 1, \cdots, d$ is the feature/dimension index.

With the result of Eq.(2.11), the objective function of Eq.(2.10) becomes

$$\sum_{i=1}^{d} \left(\frac{\widehat{w}_i^2}{\theta_i} \right) = \|\widehat{\mathbf{w}}\|_1^2. \tag{2.12}$$

As a result, optimization problem given in Eq.(2.9) is transformed into a problem identical to problem (2.5). □

Using Lemma 2.4.1, Theorem 2.3.1 can be easily proved. Eq.(2.4) in Theorem 2.3.1 comes from Eq.(2.11).

The above relationships among LASSO, nonnegative Garrote and selective ridge regression provides a probability interpretation of LASSO. To gain further insights, we can easily prove the following

Theorem 2.4.2. *The following optimization*

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\theta}} \quad & \left\| \mathbf{X}^T (\boldsymbol{\theta}^{\frac{1}{2}} \odot \mathbf{w}) - \mathbf{y} \right\|_2^2 + \lambda \|\mathbf{w}\|_2^2, \\ s.t. \quad & \boldsymbol{\theta} \geq 0, \mathbf{1}^T \boldsymbol{\theta} = h. \end{aligned} \tag{2.13}$$

where $0 < h \leq 1$ *is a constant, is identical to*

$$\min_{\widehat{\mathbf{w}}} \quad \left\| \mathbf{X}^T \widehat{\mathbf{w}} - \mathbf{y} \right\|_2^2 + \tfrac{\lambda}{h} \|\widehat{\mathbf{w}}\|_1^2 . \tag{2.14}$$

Once the optimal solution $\widehat{\mathbf{w}}^*$ *to problem (2.14) is found, optimal solution to problem (2.13) is given by*

$$\theta_i^* = \frac{h\,|\widehat{w}_i^*|}{\|\widehat{\mathbf{w}}^*\|_1}, \; w_i^* = (\theta_i^*)^{-\frac{1}{2}} \widehat{w}_i^*, \tag{2.15}$$

where $i = 1, \cdots, d$ *is the feature/dimension index. When* $\widehat{w}_i^* = 0, w_i^* = 0$*. Note that* $\mathbf{1}^T \boldsymbol{\theta}^* = h$*.*

Theorem 2.4.2 implies that when we wish to select less number of features using a smaller $h < 1$, we need to increase the regularization strength λ/h in front of squared ℓ_1-norm, see Eq.(2.14).

2.4.1 A Ranking Method

Strictly speaking, in order to use LASSO to select m features, one has to set λ appropriately to a value λ_m so that exactly m features in optimal solution $\widehat{\mathbf{w}}^*$ are nonzero. The less number of features we desire, the stronger regularization we need to apply — consistent with Theorem 2.4.2. We will call this method as strict λ_m method. This strict λ_m method is computationally expensive.

The probability derivation of LASSO of Theorems 2.3.1 and 2.4.2, as the selection vector $\boldsymbol{\theta}$ from the selective ridge regression, naturally provides a ranking scheme of the features. Once we computed the solution to the LASSO problem Eq.(2.5), from Eq.(2.11), the importance of feature i is proportional to $|\widehat{w}_i^*|$. In other words, we rank the importance of features according to $(|\widehat{w}_1^*|, \cdots, |\widehat{w}_d^*|)$, and select the top m ranked features from the sorted order. This ranking selection method is fast in practice.

These two selection methods usually lead to different selected feature sets. In our experiments and from reading many research publications by other researchers, the feature set selected from ranking method generally performs better than the feature set selected via the strict λ_m method.

A simple explanation is that the strict λ_m method usually leads to a larger λ_m as compared to the λ used in the ranking method. The larger λ_m used in LASSO usually penalizes the regression too severely and thus altered the structural relation among the features. In the ranking method, a smaller λ is used which does not alter the relation among the features. This explanation is further strengthened from the point of view of the selection vector $\boldsymbol{\theta}$ in selective ridge regression.

2.4.2 Beyond the Linear Regression Loss

In formulations Eqs.(2.1, 2.6, 2.13), the error/loss term uses linear regression. But they can be any other forms of loss. The proofs of Theorems 2.3.1 and 2.4.2 only depend on the regularization term, and thus hold without any change. In other words, the process from the ℓ_2 regularization to the $\ell_{1,2}$ regularization is purely due to the transformation of probabilistic selection.

2.5 Feature Selection Using $\ell_{1,2}$-Norm

From here on, we use $\mathbf{W}$ to replace $\widehat{\mathbf{W}}$ in Eq.(2.1) for notational simplicity.

As explained earlier, flexible feature selection does not enforce rigourously that features selected for every class are exactly same. This is naturally done in the $\ell_{1,2}$ regularization based selection we propose in this work, written explicitly here for clarity,

$$\|\mathbf{W}\|_{1,2}^2 = \sum_{j=1}^{k} \left(\sum_{i=1}^{d} |W_{ij}| \right)^2 . \tag{2.16}$$

As regularization strength parameter λ goes large, different elements in the ℓ_1 norm of j-th column of $\mathbf{W}$ (i.e. $\sum_{i=1}^{d} |W_{ij}|$) for a fixed class j compete with each other, and only a few elements (corresponding to different features) will survive (be nonzero), i.e., these features being selected for class j.

To the best of our knowledge, however, flexible feature selection has not been thoroughly investigated so far. The main trend in feature selection is using $\ell_{2,1}$-norm based formalisms [16, 17, 18, 19], enforcing joint sparsity and selecting rows of weight matrix $\mathbf{W}$.

We note that the competition and survival property explained above for $\|\mathbf{W}\|_{1,2}^2$ also happens in exclusive lasso of Zhou et al. [26]. Their formulation is different from our approach here. They use the regularization

$$\left\|\mathbf{W}^T\right\|_{1,2}^2 = \sum_{i=1}^{d}\left(\sum_{j=1}^{k}|W_{ij}|\right)^2. \tag{2.17}$$

As regularization strength parameter λ goes large, different elements in the ℓ_1 norm of i-th row of $\mathbf{W}$ (i.e. $\sum_{j=1}^{k}|W_{ij}|$) for a fixed feature i compete with each other, i.e., they are mutually exclusive, and only a few elements (corresponding to different classes) will survive (be nonzero), i.e., feature i being selected for these classes. This competition and survival property is the prominent feature of "exclusive LASSO". In Kong et al [27], they use exclusive group norm, $\sum_g \|\mathbf{w}_g\|_1^2$ (where g is the group index) which is very similar to exclusive lasso, except only 2-class case is considered there.

In summary, both our proposed $\ell_{1,2}$-norm based feature selection $\|\mathbf{W}\|_{1,2}^2$ and the exclusive LASSO $\left\|\mathbf{W}^T\right\|_{1,2}^2$ have the competition and survival property (i.e. the "exclusive" property), and can be used for flexible feature selection. However, $\ell_{2,1}$-norm based feature selection $\left\|\mathbf{W}^T\right\|_{2,1}$ is not suitable for flexible feature selection.

2.6 Efficient Algorithms

We wish to solve the $\ell_{1,2}$-norm based feature selection and the exclusive lasso (eLASSO). They are expressed as

$$E(\mathbf{W}) = \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_F^2, \tag{2.18}$$

$$J_{12}(\mathbf{W}) = E(\mathbf{W}) + \lambda\|\mathbf{W}\|_{1,2}^2, \tag{2.19}$$

$$J_{\text{eLASSO}}(\mathbf{W}) = E(\mathbf{W}) + \lambda\left\|\mathbf{W}^T\right\|_{1,2}^2. \tag{2.20}$$

We use an iterative algorithm to solve the problem.

Let $\mathbf{W}^0, \mathbf{W}^1, \cdots, \mathbf{W}^t, \mathbf{W}^{t+1}, \cdots$ be the solutions at different stages. Our task here is (A) derive an update algorithm $\mathbf{W}^{t+1} = f(\mathbf{W}^t)$, and (B) prove its convergence: $J(\mathbf{W}^{t+1}) \leq J(\mathbf{W}^t)$.

We use the auxiliary function approach widely adopted in nonnegative matrix factorization [31, 32, 33] to derive an efficient algorithm. A function $G(\mathbf{W}, \widetilde{\mathbf{W}})$ is the auxiliary function of $J(\mathbf{W})$, if it satisfies condition (C1)

$$J(\mathbf{W}) \le G(\mathbf{W}, \widetilde{\mathbf{W}}), \forall \mathbf{W}, \widetilde{\mathbf{W}},$$

and condition (C2)

$$J(\mathbf{W}) = G(\mathbf{W}, \mathbf{W}), \forall \mathbf{W}.$$

The key step is finding the auxiliary function for the objective $J_{12}(\mathbf{W})$ and $J_{\text{eLASSO}}(\mathbf{W})$. We have

Theorem 2.6.1. *An auxiliary function for $J_{12}(\mathbf{W})$ is*

$$\begin{aligned} & G_{12}(\mathbf{W}, \mathbf{W}^t) \\ = \; & E(\mathbf{W}) + \lambda \textstyle\sum_{j=1}^{k} \left(\sum_{i=1}^{d} \frac{W_{ij}^2}{|W_{ij}^t|} \right) \|\mathbf{w}_j^t\|_1 \\ = \; & E(\mathbf{W}) + \lambda \textstyle\sum_{j=1}^{k} \mathbf{w}_j^T \mathbf{D}_j \mathbf{w}_j, \end{aligned} \tag{2.21}$$

where

$$\mathbf{D}_j = \|\mathbf{w}_j^t\|_1 \text{diag}(1/|W_{1j}^t|, \cdots, 1/|W_{dj}^t|), \tag{2.22}$$

and $\mathbf{w}_j$ is a column vector. An auxiliary function for J_{eLASSO} is

$$\begin{aligned} & G_{\text{eLASSO}}(\mathbf{W}, \mathbf{W}^t) \\ = \; & E(\mathbf{W}) + \lambda \textstyle\sum_{i=1}^{d} \left(\sum_{j=1}^{k} \frac{W_{ij}^2}{|W_{ij}^t|} \right) \|(\mathbf{w}^i)^t\|_1 \\ = \; & E(\mathbf{W}) + \lambda \textstyle\sum_{i=1}^{d} \mathbf{w}^i \mathbf{H}_i (\mathbf{w}^i)^T, \end{aligned} \tag{2.23}$$

where

$$\mathbf{H}_i = \|(\mathbf{w}^i)^t\|_1 \text{diag}(1/|W_{i1}^t|, \cdots, 1/|W_{ik}^t|), \tag{2.24}$$

and $\mathbf{w}^i$ is a row vector.

The proof of this theorem is given below.

In the following, we focus on deriving the update algorithm of $\ell_{1,2}$-norm based feature selection using $J_{12}(\mathbf{W})$. Algorithm for $J_{\text{eLASSO}}(\mathbf{W})$ can be obtained in identical fashion.

2.6.1 The Update Algorithm

Using Theorem 2.6.1, the update algorithm is given by

$$\mathbf{W}^{t+1} = \arg\min_{\mathbf{W}} G_{12}(\mathbf{W}, \mathbf{W}^t). \tag{2.25}$$

This is solved by setting $\frac{\partial G_{12}(\mathbf{W},\mathbf{W}^t)}{\partial \mathbf{W}} = 0$. The solution is

$$\mathbf{w}_j^{t+1} = (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{D}_j)^{-1}(\mathbf{X}\mathbf{y}_j), \tag{2.26}$$

where $j = 1, \cdots, k$ is the class index, and $\mathbf{D}_j$ is defined in Eq.(2.22).

Eq.(2.26) is the updating equation for the j-the column of weight matrix $\mathbf{W}$. Since $G_{12}(\mathbf{W}, \mathbf{W}^t)$ is a strict convex function in $\mathbf{W}$, $\mathbf{w}_j^{t+1}$ obtained is the global optimal solution of $J_{12}(\mathbf{W})$.

This is a convergent update algorithm, because we have

$$J_{12}(\mathbf{W}^{t+1}) \leq G_{12}(\mathbf{W}^{t+1}, \mathbf{W}^t) \leq G_{12}(\mathbf{W}^t, \mathbf{W}^t) = J_{12}(\mathbf{W}^t).$$

The first inequality is due to the condition (C1) for the auxiliary function. The second inequality comes from the fact that $\mathbf{W}^{t+1}$ is the global optimal solution for Eq.(2.25). The third equality comes from auxiliary function condition (C2).

In summary, we have derived the update algorithm outlined in Algorithm 1 and proved its convergence.

Proof of Theorem 4.

Auxiliary function condition (C1):

$$J_{12}(\mathbf{W}^{t+1}) \leq G_{12}(\mathbf{W}^{t+1}, \mathbf{W}^t).$$

Let the difference between left-hand-side and right-hand-side of above inequality defined as $\Delta = \text{LHS} - \text{RHS}$.

Thus, we obtain the following

Algorithm 1 Efficient algorithm for solving the $\ell_{1,2}$-norm based feature selection.

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, labels $\mathbf{Y} \in \mathbb{R}^{n \times k}$.
Output: $\mathbf{W} \in \mathbb{R}^{d \times k}$, $\mathbf{D}_j \in \mathbb{R}^{d \times d}$, $j = 1, \cdots, k$.
Set $t = 0$.
Initialize $\mathbf{W}^t$.
repeat
    for each class $j \in \{1, \cdots, k\}$ do
        Compute $\mathbf{D}_j$ via Eq.(2.22).
        Compute $\mathbf{w}_j^{t+1}$ via Eq.(2.26).
    end for
    Set $t = t + 1$.
until Converges
```

$$
\begin{aligned}
\Delta &= \left(E(\mathbf{W}^{t+1}) + \lambda \|\mathbf{W}^{t+1}\|_1^2 \right) - \left(E(\mathbf{W}^{t+1}) + \lambda \sum_{j=1}^{k} \left(\mathbf{w}_j^{t+1}\right)^T \mathbf{D}_j \left(\mathbf{w}_j^{t+1}\right) \right) \\
&= \lambda \left(\sum_{j=1}^{k} \left\|\mathbf{w}_j^{t+1}\right\|_1^2 \right) - \lambda \left(\sum_{j=1}^{k} \left(\mathbf{w}_j^{t+1}\right)^T \mathbf{D}_j \left(\mathbf{w}_j^{t+1}\right) \right) \\
&= \lambda \sum_{j=1}^{k} \left[\left(\sum_{i=1}^{d} \left|W_{ij}^{t+1}\right| \right)^2 - \left(\sum_{i=1}^{d} \frac{\left|W_{ij}^{t+1}\right|^2}{\left|W_{ij}^{t}\right|} \right) \left(\sum_{i=1}^{d} \left|W_{ij}^{t}\right| \right) \right] \\
&= \lambda \sum_{j=1}^{k} \left[\left(\sum_{i=1}^{d} A_{ij} B_{ij} \right)^2 - \left(\sum_{i=1}^{d} A_{ij}^2 \right) \left(\sum_{i=1}^{d} B_{ij}^2 \right) \right] \leq 0
\end{aligned}
\tag{2.27}
$$

where $A_{ij} = \frac{\left|W_{ij}^{t+1}\right|}{\sqrt{\left|W_{ij}^{t}\right|}}$, $B_{ij} = \sqrt{\left|W_{ij}^{t}\right|}$. The last inequality in Eq.(2.27) is obtained according to the Cauchy-Schwarz[1] inequality, which proves condition (C1).

[1] Given any two vectors $\mathbf{x}$ and $\mathbf{y}$, the Cauchy-Schwarz inequality states, in the inner product space, it is always true that $(\sum_i x_i y_i)^2 \leq (\sum_i x_i^2)(\sum_i y_i^2)$.

Auxiliary function condition (C2):

$$G_{12}(\mathbf{W}^t, \mathbf{W}^t) = J_{12}(\mathbf{W}^t).$$

From the Eq.(2.21), we obtain the following

$$\begin{aligned} & G_{12}(\mathbf{W}^t, \mathbf{W}^t) \\ = \ & E(\mathbf{W}^t) + \lambda \sum_{j=1}^{k} \left(\sum_{i=1}^{d} \frac{(W_{ij}^t)^2}{|W_{ij}^t|} \right) \|\mathbf{w}_j^t\|_1 \\ = \ & E(\mathbf{W}^t) + \lambda \sum_{j=1}^{k} \|\mathbf{w}_j^t\|_1^2 \\ = \ & J_{12}(\mathbf{W}^t), \end{aligned} \tag{2.28}$$

which proves condition (C2). Thus, Theorem 2.6.1 is proved. □

During the computation, many of the elements W_{ij} become zero due to sparsity. We therefore replace $1/|W_{ij}|$ by $1/(|W_{ij}| + \epsilon)$, where ϵ is a small number $1e{-}7$.

2.7 Experiment

For purpose of verifying the effectiveness of our flexible feature selection method via $\ell_{1,2}$-norm, extensive experiments on six benchmark datasets are conducted in comparison with six state-of-the-art algorithms.

2.7.1 Description of Benchmark Datasets

In our experiments, six benchmark datasets including image and bio-microarray data are used to study the performance of feature selection methods on multi-class classification. The description of all the datasets are given as follows.

Image dataset: there are three image datasets, including MNIST[2] [34], BinAlpha[3], AT&T[4]. Each instance is represented by a vector with all the pixel values in

[2] In MNIST, one hundred samples are randomly chosen out of each class to form a smaller dataset in our experiments.

[3] `https://cs.nyu.edu/~roweis/data.html`

[4] `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`

an image. In MNIST, handwritten digits from 0 to 9 are collected as samples. On the other hand, samples in BinAlpha are composed of handwritten letters from A to Z. Both digits and letters have been size-normalized and centered in a fixed-size image. AT&T, also known as the ORL database of faces, is the widely used face recognition dataset, in which images were taken at different times varying the lighting, facial expressions, and facial details.

Microarray dataset: there are three microarray datasets, including Carcinomas [35, 36], Lung [37], TOX[5] [38]. Each instance is represented by a vector with all the genes expression values. Under the first-generation molecular classification scheme, both Carcinomas and Lung are constructed to identify gene subsets whose expression typifies each cancer class, and quantify the extent to which genes are related to specific tumor type. In another hand, TOX focuses on discovering the time-course of changes in adipocyte morphology, adipokines and the global transcriptional landscape in visceral white adipose tissue, during the development of diet-induced obesity.

As compared to image dataset, microarray dataset usually involves a relatively small number of data instances but following with a extremely high dimension of features.

The detail of benchmark datasets is summarized in Table 2.1.

2.7.2 Classification Result and Analysis

Baseline methods: our $\ell_{1,2}$-norm based flexible feature selection method is compared to six state-of-the-art algorithms, including feature selection via $\ell_{2,1}$-norm [16, 17, 18], feature selection via $\ell_{1,\infty}$-norm [21], exclusive lasso (eLASSO) [26], mRMR [12], F-statistic [10], ReliefF [11]. Towards a fair comparison, the hyperparameter λ

[5] `http://featureselection.asu.edu/datasets.php`

Dataset	#Classes	#Instances	#Features
MNIST	10	1000	784
BinAlpha	26	1014	320
AT&T	40	400	644
Carcinomas	11	174	9182
Lung	5	203	3312
TOX	4	171	5748

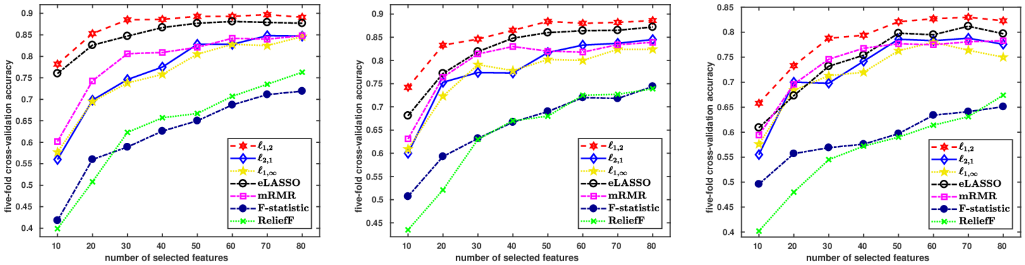
Table 2.1: Summary descriptions of dataset.

in sparse coding based models, such as $\ell_{1,2}$, $\ell_{2,1}$, $\ell_{1,\infty}$, is adjusted to achieve the same number of nonzero elements in the weight matrix $\mathbf{W}$.

Classifiers: k-nearest neighbor (KNN), support vector machine (SVM), and linear regression (LR) with five-fold cross validation are used to evaluate the performance of feature selection on classification. The average of classification performance on different five folds are reported as the final accuracy. The parameter k in KNN is set as 3. LIBSVM [39] is used as practical implementation of SVM, in which the kernel is set as linear and $C = 1$.

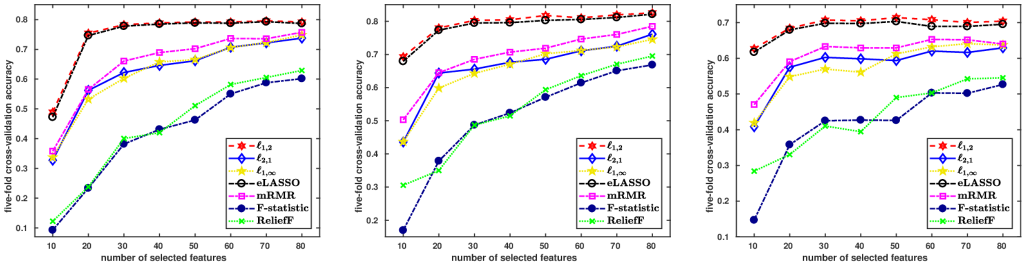
Analysis of experimental results: As it can be seen in Fig. 2.1−2.6 that the classification using aforementioned seven feature selection methods is performed on six benchmark datasets. From left to right in each figure, classifiers that we used are KNN, SVM, and LR respectively. The number of selected features for each method ranges from 10 to 80, which is marked as the scale of x-axis. The y-axis shows the averaged accuracy of five-fold cross validation.

Among these baseline methods, the simplest F-statistic has the worst performance overall. As compared to F-statistic, another two filter-type methods, such as mRMR and ReliefF, improve the classification accuracy greatly. Moreover, mRMR can even beat sparse coding based methods for example $\ell_{2,1}$-norm and $\ell_{1,\infty}$-norm in some cases.



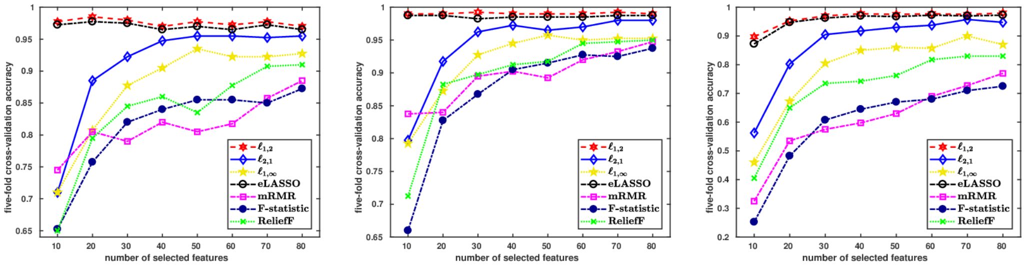
(a) KNN (b) SVM (c) LR

Figure 2.1: $\ell_{1,2}$ versus state-of-the-arts on MNIST dataset.



(a) KNN (b) SVM (c) LR

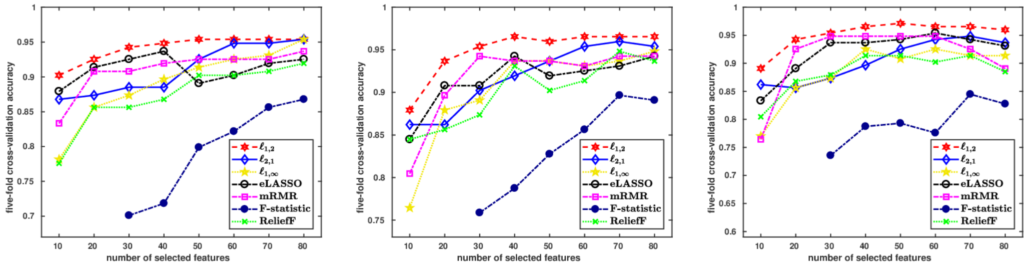
Figure 2.2: $\ell_{1,2}$ versus state-of-the-arts on BinAlpha dataset.



(a) KNN (b) SVM (c) LR

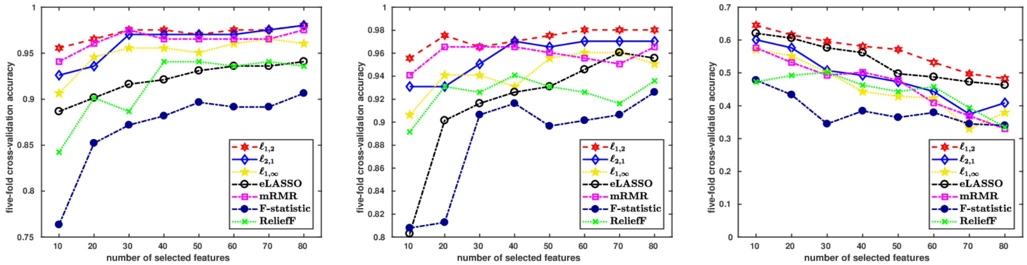
Figure 2.3: $\ell_{1,2}$ versus state-of-the-arts on AT&T dataset.

However, filter-type methods are inferior to sparse coding based methods in general. Feature selection via $\ell_{2,1}$-norm performs very close to feature selection via $\ell_{1,\infty}$-norm when classifying not only images but also bio-microarray data, since both methods share the same property that aims at searching a subset of class-shared features across all the data instances. Only the results obtained on AT&T dataset, $\ell_{2,1}$ is obviously better than $\ell_{1,\infty}$ around 5.0%.



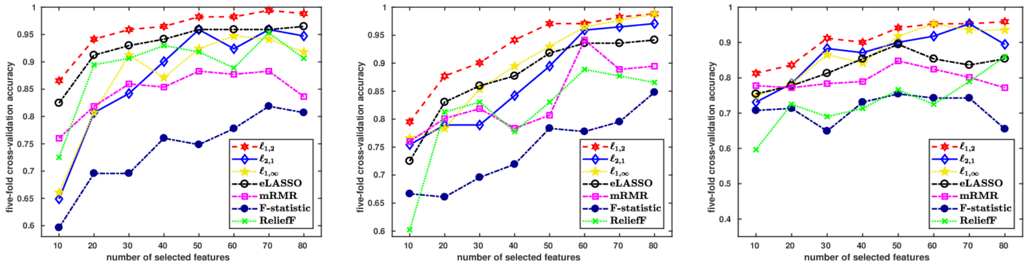
(a) KNN (b) SVM (c) LR

Figure 2.4: $\ell_{1,2}$ versus state-of-the-arts on Carcinomas dataset. F-statistic using top 10 and 20 features is not plotted in the figure, since the classification accuracy is way below the scale of y-axis.



(a) KNN (b) SVM (c) LR

Figure 2.5: $\ell_{1,2}$ versus state-of-the-arts on Lung dataset.



(a) KNN (b) SVM (c) LR

Figure 2.6: $\ell_{1,2}$ versus state-of-the-arts on TOX dataset.

Among sparse coding based methods, eLASSO is an outstanding one which selects exclusive features as the main purpose, only performing slightly lower than our $\ell_{1,2}$-norm based method around 1.0% on BinAlpha and AT&T. Nevertheless, when the dimension of features becomes very large, eLASSO has a relatively bad results on microarray datasets.

Most importantly, our flexible feature selection method via $\ell_{1,2}$-norm achieves the best results on all the six benchmark datasets as compared to state-of-the-arts. No matter which classifier is used here, $\ell_{1,2}$ has an overwhelmed advantage over six baseline methods. Besides, $\ell_{1,2}$ has a stable performance without huge degradation, when using any number of feature subsets. Contrarily, most of baseline methods have a deteriorated performance in different degrees, when the number of selected features is relatively small. However, $\ell_{1,2}$ is better than others around 5%-10% using top 10 or 20 features. In summary, experimental results on benchmark datasets verify that $\ell_{1,2}$ based flexible feature selection is a more nature way to measure the importance of features for each class than class-shared selections.

2.8 Conclusion

In this work, we derive LASSO and $\ell_{1,2}$-norm feature selection from a probabilistic framework. In addition, we further propose a feature selection approach based on $\ell_{1,2}$-norm, allowing certain flexibility that selected features do not have to be exactly same for all classes. The resulting features lead to significantly better classification than state-of-the-art methods on six benchmark datasets, including images and bio-microarray data.

CHAPTER 3

ROBUST FLEXIBLE FEATURE SELECTION VIA EXCLUSIVE L21 REGULARIZATION

3.1 Introduction

Feature selection plays an important role in many machine learning tasks. The main purpose is to remove irrelevant and redundant noise features in high-dimensional data space. The selected features will help to reduce the computation cost and improve the performance on real-world applications.

There are many research works on feature selection over the years. Generally, feature selection methods can be divided into three main categories [9]: wrapper method, filter method, and sparse coding based method (also known as embedded method). The most representative wrapper method is support vector machine recursive feature elimination (SVM-RFE) [13], but the computation cost is extremely high. Contrarily, filter method is very efficient such as F-statistic [10], ReliefF [11], minimum redundancy maximum relevance (mRMR) [12].

Recently, sparse coding based methods have been widely investigated by researchers, and applied to the study of feature selections. Classic least absolute shrinkage and selection operator (LASSO) [14] is a regression based analysis method that incurs the sparsity on weights via ℓ_1-norm. ℓ_1-SVM [15] and hybrid huberized SVM (HHSVM) [40] are introduced to further improve performance on two-class problem. LASSO can be derived from probabilistic selection on ridge regression [41].

Towards solving multi-class problem, researchers start to search a subset of features shared by all the classes, also known as multi-task feature learning (MTFL).

$\ell_{2,1}$-norm is the most widely used regularization, developed in [16, 17, 18, 19]. During the same period, Quattoni et al of [21] propose the $\ell_{1,\infty}$-norm regularization, which shares the similar property of row sparsity as $\ell_{2,1}$-norm. As compared to class-shared feature selection methods, exclusive lasso (eLASSO) [26, 28] proposes to capture the negative correlation among different classes via $\ell_{1,2}$-norm, which is first introduced in [25] called as composite absolute penalties (CAP). In exclusive learning, discriminative feature is selected for each class to provide certain flexibility. Based on this, Kong et al of [27] propose to solve the mix of ℓ_1-norm and $\ell_{1,2}$-norm, for purpose of minimizing the feature correlation.

Motivated by previous works, we introduce a novel regularization called "exclusive $\ell_{2,1}$", which is short for "$\ell_{2,1}$ with exclusive lasso". The "exclusive $\ell_{2,1}$" regularization brings out joint sparsity at inter-group level and exclusive sparsity at intra-group level simultaneously. Thus, this proposed regularization can combine the advantages from different sparsity-induced penalties, which not only removes irrelevant noise features (i.e. increase the robustness via $\ell_{2,1}$-norm regularization) but also selects discriminative features for each class (i.e. provide the flexibility via $\ell_{1,2}$-norm regularization). As a result, "exclusive $\ell_{2,1}$" successfully resolves the problems for using $\ell_{2,1}$-norm alone or using $\ell_{1,2}$-norm alone.

The main contribution of this work includes: (i) a novel "exclusive $\ell_{2,1}$" regularization is proposed to conduct robust flexible feature selection; (ii) we point out some interesting properties of $\|\mathbf{w}\|_1^2$ regularization as compared to $\|\mathbf{w}\|_1$ regularization; (iii) a sorting based explicit approach is introduced to solve the $\ell_{1,2}$-norm regularization with analytic solution; (iv) an efficient augmented Lagrange multipliers (ALM) based optimization algorithm is proposed to iteratively solve the "exclusive $\ell_{2,1}$" regularization in a row-wise fashion; (v) experimental results on twelve benchmark datasets demonstrate that the proposed regularization outperforms state-of-the-arts.

3.2 Notations and Definitions

Throughout this chapter, scalars, vectors, and matrices are denoted as lower-case/capital letters, boldface lower-case letters, and boldface capital letters, respectively.

The i-th element of vector $\mathbf{w}$ is represented by w_i. Given a matrix $\mathbf{W} = (W_{ij}) \in \mathbb{R}^{d \times k}$, the i-th row is represented by $\mathbf{w}^i$ (i.e. $\mathbf{W} = [\mathbf{w}^1; \cdots; \mathbf{w}^d]$), and the j-th column is represented by $\mathbf{w}_j$ (i.e. $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_k]$). The Frobenius norm of $\mathbf{W}$ is $\|\mathbf{W}\|_F = \sqrt{\sum_{i=1}^d \sum_{j=1}^k W_{ij}^2}$. $\ell_{2,1}$-norm of $\mathbf{W}$ is $\|\mathbf{W}\|_{2,1} = \sum_{i=1}^d \|\mathbf{w}^i\|_2 = \sum_{i=1}^d \big(\sum_{j=1}^k W_{ij}^2 \big)^{1/2}$. $\ell_{1,2}$-norm of $\mathbf{W}$ is $\|\mathbf{W}\|_{1,2}^2 = \sum_{i=1}^d \|\mathbf{w}^i\|_1^2 = \sum_{i=1}^d \big(\sum_{j=1}^k |W_{ij}| \big)^2$.

$\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ represents n data points, where $\mathbf{x}_i \in \mathbb{R}^d$, and the corresponding class labels are defined as $\mathbf{Y} = [\mathbf{y}^1; \cdots; \mathbf{y}^n] \in \mathbb{R}^{n \times k}$, where $\mathbf{y}^i \in \mathbb{R}^k$ is one-hot encoding vector and $y_j^i = 1$ or $Y_{ij} = 1$ means i-th sample belonging to j-th class.

3.3 Exclusive $\ell_{2,1}$ Regularization

In general, sparse coding based method can be formulated as

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k}} \; f(\mathbf{W}) + \lambda \Omega(\mathbf{W}), \tag{3.1}$$

where $f(\mathbf{W})$ is the loss term to measure the difference/error between ground truth (i.e. given class labels) and prediction, $\Omega(\mathbf{W})$ is the sparsity-induced regularization term, and λ is the hyperparameter to control the level of sparsity in $\mathbf{W}$.

Our work is motivated from the following observations. The $\ell_{2,1}$ norm based feature selection:

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times k}} \; f(\mathbf{W}) + \lambda \|\mathbf{W}\|_{2,1} \tag{3.2}$$

incurs joint sparsity on rows (i.e. all the elements in a row are shrunk to zero values). A selected non-zero row could still have some elements with small (in magnitude) numerical values. Suppose one of them is W_{ij}. This implies i-feature is not highly correlated with j-th class. For example, in object recognition, features related to wheel should be selected for automobiles such as car, bus, truck, etc, but should not be selected for animals such as bird, cat, horse, etc. Thus $\ell_{2,1}$ alone is too rigid for feature selection.

On the other end, exclusive lasso using $\ell_{1,2}$-norm

$$\min_{\mathbf{W}\in\mathbb{R}^{d\times k}} \; f(\mathbf{W}) + \lambda\|\mathbf{W}\|_{1,2}^2 \tag{3.3}$$

selects discriminative features for each class. Here, as λ increases, different elements in squared ℓ_1-norm of i-th row $\mathbf{w}^i$ are competing with each other to survive. Thus, at least one element in row $\mathbf{w}^i$ survive (remaining non-zero), as regularization strength increases to a certain value. This is so-called "exclusive sparsity". The problem with exclusive lasso in this context is: all features will be selected, because for each feature i, there will be some non-zero elements even at large regularization strength. For example, in gene-expression analysis of smoking and non-smoking, each gene will be selected at least for one case using $\ell_{1,2}$-norm. As a results, those irrelevant noise genes will not be screened out. This outcome is not what we desire, since we only want to find out a few significant genes which are highly correlated with either smoking or non-smoking. Thus, $\ell_{1,2}$ along is not robust for feature selection.

Towards resolving above main concerns for using $\ell_{2,1}$ regularization alone or using exclusive lasso alone, we propose to combine them together as a new regularization defined as

$$\Omega(\mathbf{W}) = \alpha\,\|\mathbf{W}\|_{2,1} + \beta\,\|\mathbf{W}\|_{1,2}^2\,, \tag{3.4}$$

which is called as "exclusive $\ell_{2,1}$", short for "$\ell_{2,1}$ with exclusive lasso".

How two different type of sparsity-induced norms can work synergistically to achieve the robust flexible feature selection goal?

(i) As the regularization strength α increases, $\ell_{2,1}$-norm enforces more and more rows to zero, which thus helps $\ell_{1,2}$-norm to eliminate those irrelevant noise rows. This resolves the concern with exclusive lasso alone regularization.

(ii) As the regularization strength β increases, $\ell_{1,2}$-norm enforces each row to have at least one nonzero, which thus helps $\ell_{2,1}$-norm to eliminate small (in magnitude) nonzero elements in nonzero row. This resolves the concern with $\ell_{2,1}$ alone regularization.

3.3.1 An Illustration

We give an illustration here to describe the difference between above-mentioned three sparsity-induced regularizations.

The synthetic data $\mathbf{X}$, $\mathbf{Y}$ is given in Eq. (3.5), where $d = 7$, $n = 8$, $k = 3$.

$$\mathbf{X}^T = \begin{bmatrix} 0.463 & 0.319 & -0.100 & 0.526 & 0.535 & 0.329 & 0.475 \\ 0.296 & 0.192 & 0.058 & -0.076 & 0.152 & 0.313 & -0.114 \\ 0.196 & 0.189 & 0.167 & -0.280 & 0.267 & -0.246 & 0.164 \\ 0.330 & 0.357 & 0.027 & -0.001 & 0.118 & 0.058 & 0.191 \\ 0.332 & 0.035 & -0.002 & 0.280 & 0.111 & -0.043 & 0.104 \\ -0.022 & -0.026 & 0.770 & 0.189 & 0.196 & -0.146 & -0.121 \\ -0.217 & 0.028 & 0.404 & 0.359 & 0.335 & -0.282 & -0.235 \\ 0.396 & 0.297 & 0.260 & 0.241 & 0.193 & 0.038 & 0.101 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

The loss function we used for this illustration is defined as

$$f(\mathbf{W}) = \|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\|_F^2, \tag{3.6}$$

which is the standard least square loss. The hyperparameter in front of different norms is adjusted to obtain the same level of sparsity in $\mathbf{W}$. The learned matrices are given in Eqs. (3.7, 3.8, 3.9), where the number of non-zero elements in the weight matrix $\mathbf{W}$ is enforced to 12 for each regularization. Their differences are explained as follows:

(i) $\mathbf{W}_{21}$ ($\ell_{2,1}$-*norm*): a feature can be selected by all the classes (e.g. *3rd* row is selected for *1st*, *2nd*, and *3rd* classes), or can be eliminated (e.g. *4th* row is a zero vector).

$$\mathbf{W}_{21} = \begin{bmatrix} 0.764 & 0.587 & 0.378 \\ 0.097 & 0.033 & 0.082 \\ 0.054 & 0.531 & 1.003 \\ \textbf{-0.000} & \textbf{0.000} & \textbf{0.000} \\ 0.151 & 0.030 & 0.126 \\ \textbf{0.000} & \textbf{0.000} & \textbf{-0.000} \\ \textbf{0.000} & \textbf{-0.000} & \textbf{0.000} \end{bmatrix} \tag{3.7}$$

(ii) $\mathbf{W}_{12}$ (*exclusive LASSO*): a feature can be selected by some classes (e.g. *5th* row is selected for *1st* and *3rd* classes; *6th* row is selected only for *3rd* class), but can not be eliminated since the weight matrix has no zero rows.

$$\mathbf{W}_{12} = \begin{bmatrix} 0.336 & 0.352 & \mathbf{0.000} \\ 0.287 & \mathbf{0.000} & 0.358 \\ \mathbf{0.000} & 0.070 & 0.758 \\ -0.009 & 0.173 & \mathbf{0.000} \\ 0.326 & \mathbf{0.000} & 0.298 \\ \mathbf{0.000} & \mathbf{0.000} & -0.344 \\ 0.333 & \textbf{-0.000} & \mathbf{0.000} \end{bmatrix} \tag{3.8}$$

(iii) $\mathbf{W}_{\text{exL21}}$ (*the proposed "exclusive $\ell_{2,1}$" regularization*): a feature can be selected by all the classes (e.g. *2nd* row is selected for *1st*, *2nd*, and *3rd* classes), or can be selected by some classes (e.g. *3rd* row is selected for *2nd* and *3rd* classes), or can be eliminated (e.g. *4th* row is a zero vector).

$$\mathbf{W}_{\text{exL21}} = \begin{bmatrix} 0.192 & 0.114 & \mathbf{0.000} \\ 0.132 & 0.014 & 0.090 \\ \mathbf{0.000} & 0.041 & 0.358 \\ \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.000} \\ 0.133 & \mathbf{-0.000} & 0.137 \\ 0.017 & 0.004 & -0.024 \\ \mathbf{0.000} & \mathbf{0.000} & \mathbf{0.000} \end{bmatrix} \tag{3.9}$$

Based on learned weight matrices in Eqs. (3.7, 3.8, 3.9), we summarize the features selected by different approaches in Tables 3.1, 3.2, 3.3, where "✓" represents one feature is selected for certain class and "✗" represents one feature is not selected for certain class.

Thus it can be seen that the proposed exclusive $\ell_{2,1}$ regularization combines the advantages of different sparsity-induced regularizations, which not only removes irrelevant noise features (i.e. increase the robustness via $\ell_{2,1}$-norm), but also selects discriminative features for each class (i.e. provide the flexibility via $\ell_{1,2}$-norm).

3.4 Understanding the Exclusive Sparsity of $\ell_{1,2}$-Norm

3.4.1 Interesting Property of $\|\mathbf{w}\|_1^2$ Regularization

In this work we use $\|\mathbf{w}\|_1^2$ regularization for flexible feature selection. Here we point out some interesting properties of this regularization.

Consider $\|\mathbf{w}\|_1^2$ regularization first. We investigate the following simple proximal operator-type problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}-\mathbf{a}\|_2^2 + \lambda\|\mathbf{w}\|_1^2. \tag{3.10}$$

Dataset	class-1	class-2	class-3
feature-1	✓	✓	✓
feature-2	✓	✓	✓
feature-3	✓	✓	✓
feature-4	✗	✗	✗
feature-5	✓	✓	✓
feature-6	✗	✗	✗
feature-7	✗	✗	✗

Table 3.1: The relation between feature and class in $\ell_{2,1}$ regularization.

Dataset	class-1	class-2	class-3
feature-1	✓	✓	✗
feature-2	✓	✗	✓
feature-3	✗	✓	✓
feature-4	✓	✓	✗
feature-5	✓	✗	✓
feature-6	✗	✗	✓
feature-7	✓	✗	✗

Table 3.2: The relation between feature and class in $\ell_{1,2}$ regularization.

Dataset	class-1	class-2	class-3
feature-1	✓	✓	✗
feature-2	✓	✓	✓
feature-3	✗	✓	✓
feature-4	✗	✗	✗
feature-5	✓	✗	✓
feature-6	✓	✓	✓
feature-7	✗	✗	✗

Table 3.3: The relation between feature and class in exclusive $\ell_{2,1}$ regularization.

This is very similar to the standard ℓ_1-norm regularization problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{w}-\mathbf{a}\|_2^2+\lambda\|\mathbf{w}\|_1. \tag{3.11}$$

which has been thoroughly studied in connection to lasso [14].

There exits a widely held belief that optimization problems Eq. (3.10) and Eq. (3.11) behave very similarly and their solutions have identical sparsity pattern.

This belief come from the following reasoning. Problem (3.10) is equivalent to

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d} \quad & \|\mathbf{w}-\mathbf{a}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{w}\|_1^2\leq t, \end{aligned} \tag{3.12}$$

for some parameter t. And problem (3.11) is equivalent to

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^d} \quad & \|\mathbf{w}-\mathbf{a}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{w}\|_1\leq t, \end{aligned} \tag{3.13}$$

for some parameter t.

However, this widely held belief is incorrect.

Let $\mathbf{w}^*_{\ell_{12}}$ be the optimal solution for problem (3.10). Let $\mathbf{w}^*_{\ell_1}$ be the optimal solution for problem (3.11). We illustrate their significant differences in the following two simple cases:

Case 1 is a simple problem in 2-dim. $\mathbf{a}=(2,1)$. Optimal solutions are (computed using algorithm explained later[1])

$$\begin{aligned}
&\lambda=0.1, && \mathbf{w}^*_{\ell_1}=(1.95,0.95), && \mathbf{w}^*_{\ell_{12}}=(1.75,0.75).\\
&\lambda=1, && \mathbf{w}^*_{\ell_1}=(1.5,0.5), && \mathbf{w}^*_{\ell_{12}}=(1,\mathbf{0}).\\
&\lambda=10, && \mathbf{w}^*_{\ell_1}=(\mathbf{0};\mathbf{0}), && \mathbf{w}^*_{\ell_{12}}=(0.1818;\mathbf{0}).\\
&\lambda=1000, && \mathbf{w}^*_{\ell_1}=(\mathbf{0};\mathbf{0}), && \mathbf{w}^*_{\ell_{12}}=(0.0020;\mathbf{0}).
\end{aligned}$$

[1] For standard $\|\mathbf{w}\|_1$ regularization, Eq. (3.11) has the closed-form solution as $\mathbf{w}^*_{\ell_1}=\text{sign}(\mathbf{a})\odot[|\mathbf{a}|-\lambda/2]_+$. For $\|\mathbf{w}\|_1^2$ regularization, we propose a sorting based explicit approach (see Theorem 3.4.5) to solve Eq. (3.10), and the optimal solution $\mathbf{w}^*_{\ell_{12}}$ is given in Eq. (3.17).

Clearly as λ increases above 1, $\mathbf{w}^*_{\ell_1}$ is all zeros, but $\mathbf{w}^*_{\ell_{12}}$ has non-zero component.

Case 2. Consider the dimension is one with $\mathbf{a} = 1$. These problems can be solved analytically. The solutions are

$$\mathbf{w}^*_{\ell_1} = \left[1 - \frac{\lambda}{2}\right]_+, \ \mathbf{w}^*_{\ell_{12}} = \frac{1}{1+\lambda}.$$

Clearly, when $\lambda > 2$, $\mathbf{w}^*_{\ell_1} = 0$, but $\mathbf{w}^*_{\ell_{12}}$ is never zero not matter how large λ is.

These two cases show that as λ increases to large values, $w^*_{\ell_1}$ will become exact zero for all components; while $w^*_{\ell_{12}}$ will become zero for $d-1$ components and one component approaches $\frac{1}{1+\lambda}$ asymptotically.

3.4.2 Solving $\ell_{1,2}$-Norm Regularization

Zhou et al of [26] illustrate the sparsity of $\ell_{1,2}$-norm from a projection point of view, then solve a min-max optimization problem. Kong et al of [27] use an iteratively re-weighted method to solve $\ell_{1,2}$-norm regularization, which needs to compute the matrix inverse at each iteration.

However, both methods are not efficient, especially in high-dimensional data space. Inspired by non-negative shrinkage thresholding operator [42], we introduce a sorting based explicit approach to solve the $\ell_{1,2}$-norm regularization, which can work efficiently with augmented Lagrangian multiplier based optimization algorithm. Here, we focus on solving the simplified formulation of $\ell_{1,2}$-norm, i.e., the proximal operator defined in Eq. (3.10), which then can be applied to solve multi-class classification problem (3.22) explained later in section 3.5.

Lemma 3.4.1. *The optimal solution $\mathbf{w}^*$ of Eq. (3.10) has the following property of its sign: for $i = 1, \cdots, d$, (i) if $a_i = 0$, $w^*_i = 0$; (ii) if $a_i \neq 0$, $\text{sign}(w^*_i) = \text{sign}(a_i)$.*

Proof of Lemma 3.4.1. If $a_i = 0$, $w^*_i = 0$ can be easily verified, since $w^*_i = 0$ gives the lower objective values of Eq. (3.10) as compared to $w^*_i \neq 0$.

If $a_i \neq 0$, suppose the optimal solution is $\widehat{\mathbf{w}} = (\cdots, c, \cdots)$, where $\widehat{w}_i = c$ and $\text{sign}(c) \neq \text{sign}(a_i)$.

However, we can always find a better solution $\widetilde{\mathbf{w}} = (\cdots, -c, \cdots)$, where $\widetilde{w}_i = -c$, to decrease the value of objective function given in Eq. (3.10), since we have the following inequality

$$(\widetilde{w}_i - a_i)^2 = (|c| - |a_i|)^2 < (|c| + |a_i|)^2 = (\widehat{w}_i - a_i)^2,$$

and $|\widehat{w}_i| = |\widetilde{w}_i| = |c|$.

Thus, we have the following inequality

$$\|\widehat{\mathbf{w}} - \mathbf{a}\|_2^2 + \lambda\|\widehat{\mathbf{w}}\|_1^2 > \|\widetilde{\mathbf{w}} - \mathbf{a}\|_2^2 + \lambda\|\widetilde{\mathbf{w}}\|_1^2,$$

which leads to the conclusion that $\text{sign}(w_i^*) = \text{sign}(a_i)$. □

Lemma 3.4.2. *The optimal solution $\mathbf{w}^*$ of Eq. (3.10) has the following property of its magnitude that for $i = 1, \cdots, d$:*

$$|w_i| - |a_i| + \lambda\|\mathbf{w}\|_1 = 0, \ \textit{if } |w_i| > 0; \tag{3.14}$$

$$-|a_i| + \lambda\|\mathbf{w}\|_1\xi_i = 0, \ \xi_i \in [0, 1], \ \textit{if } |w_i| = 0, \tag{3.15}$$

where $\xi_i \in [0, 1]$ is the subgradient of function $f(x) = |x|, x \geq 0$ at $x = 0$.

Proof of Lemma 3.4.2. Eq. (3.10) can be rewritten as

$$\min_{\mathbf{w} \in \mathbb{R}^d} J(\mathbf{w}) = \sum_{i=1}^{d} (|w_i| - |a_i|)^2 + \lambda \left(\sum_{i=1}^{d} |w_i| \right)^2 \tag{3.16}$$

since $[\text{sign}(w_i)]^2 = [\text{sign}(a_i)]^2 = 1$, according to Lemma 3.4.1,

Taking the derivative of $J(\mathbf{w})$ w.r.t $|w_i|$ and setting $\frac{\partial J(\mathbf{w})}{\partial |w_i|} = 0$, we have the same first-order optimality conditions defined in Eq. (3.14) and Eq. (3.15). □

Proposition 3.4.3. *As λ increases to large values, at least one element w_i in $\mathbf{w}$ will survive (i.e. remaining non-zero, $|w_i| > 0$), given $\mathbf{a} \neq \mathbf{0}$. Otherwise, $\mathbf{w} = \mathbf{0}$ will lead to $\mathbf{a} = \mathbf{0}$ according to Eq. (3.15).*

Definition 3.4.4. *Given* $\mathbf{a} = [a_1, \cdots, a_d] \in \mathbb{R}^d$*,* $\mathcal{S}$ *denotes a d-dimensional vector with* $\mathcal{S}_i \neq \mathcal{S}_j$*,* $\bigcup_{i=1}^{d} \mathcal{S}_i = \{1, \cdots, d\}$*, and each element* $\mathcal{S}_i$ *represents the indexes of the descending order with respect to* $\mathbf{a}$*, such as* $|a_{\mathcal{S}_1}| \geq |a_{\mathcal{S}_2}| \geq \cdots \geq |a_{\mathcal{S}_d}|$*.*

Theorem 3.4.5. *Given the d-dimensional vector* $\mathcal{S}$ *with respect to* $\mathbf{a}$*, the optimal solution of Eq. (3.10) is given by*

$$\mathbf{w}^* = \operatorname{sign}(\mathbf{a}) \odot \left[|\mathbf{a}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau \right]_+, \tag{3.17}$$

where $\odot$ *is the Hadamard product, i.e.* $[\mathbf{x} \odot \mathbf{y}]_i = x_i y_i$*,* $[\cdot]_+ = \max(\cdot, 0)$*,* $\mu_\tau = \frac{1}{\tau}\sum_{i=1}^{\tau} |a_{\mathcal{S}_i}|$*, and* τ *is the largest coordinate of* $\mathcal{S}$ *satisfying* $|a_{\mathcal{S}_\tau}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$*.*

Proof of Theorem 3.4.5. Suppose $w_{\mathcal{S}_1}$, $w_{\mathcal{S}_2}$, $\cdots$, $w_{\mathcal{S}_\tau}$ are non-zeros. By adding Eq. (3.14) for $\mathcal{S}_1$, $\mathcal{S}_2$, $\cdots$, $\mathcal{S}_\tau$ (i.e. the first τ indexes saved in $\mathcal{S}$), we have

$$\sum_{i=1}^{\tau} |w_{\mathcal{S}_i}| - \sum_{i=1}^{\tau} |a_{\mathcal{S}_i}| + \lambda\tau \|\mathbf{w}\|_1 = 0, \tag{3.18}$$

which can be equivalently rewritten as

$$\|\mathbf{w}\|_1 = \frac{\tau}{1+\lambda\tau}\mu_\tau, \tag{3.19}$$

where $\mu_\tau = \frac{1}{\tau}\sum_{i=1}^{\tau} |a_{\mathcal{S}_i}|$.

Thus, Lemma 3.4.2 and Eq. (3.19) give the optimal solution $\mathbf{w}^*$ w.r.t its magnitude as follows

$$|w^*_{\mathcal{S}_i}| = |a_{\mathcal{S}_i}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0, \text{ for } i = 1, \cdots, \tau, \tag{3.20}$$

$$|w^*_{\mathcal{S}_i}| = 0, \text{ for } i = \tau+1, \cdots, d, \tag{3.21}$$

which is equivalent to the definition of $\mathbf{w}^*$ in Eq. (3.17), since $w^*_{\mathcal{S}_i} = \operatorname{sign}(w^*_{\mathcal{S}_i})|w^*_{\mathcal{S}_i}| = \operatorname{sign}(a_{\mathcal{S}_i})|w^*_{\mathcal{S}_i}|$. □

Corollary 3.4.6. *Suppose* $j = \underset{i=1,\cdots,d}{\arg\max} |a_j|$*. As regularization strength* λ *increases to large values, the optimal solution* $\mathbf{w}^*$ *of Eq. (3.10) is given by: (i)* $w^*_i = 0$*, if* $i \neq j$*; (ii)* $w^*_i = \frac{1}{1+\lambda}a_i$*, if* $i = j$*.*

Proof of Corollary 3.4.6. As regularization strength λ increases to large values, only w_j^* with the largest coefficient $|a_j|$ will survive (i.e. remaining non-zero, $|w_j^*| > 0$), while other w_i^* $(i \neq j)$ will be equal 0, according to Proposition 3.4.3. Thus, we have $\tau = 1$ and $\mu_\tau = |a_j|$, which leads to

$$\begin{aligned} w_j^* &= \text{sign}(a_j)\left[|a_j| - \tfrac{\lambda \cdot 1}{1+\lambda \cdot 1}|a_j|\right] \\ &= \text{sign}(a_j)\tfrac{1}{1+\lambda}|a_j| \\ &= \tfrac{1}{1+\lambda}a_j, \end{aligned}$$

which completes the proof. □

As it can be seen that, w_j^* approaches to $\frac{1}{1+\lambda}$ asymptotically as λ increase to large values, which once again verifies the interesting property of $\|\mathbf{w}\|_1^2$ given in section 3.4.1.

Next, we prove the optimality of $\mathbf{w}^*$ given in Theorem 3.4.5.

Theorem 3.4.7. *When τ is the largest coordinate of $\mathcal{S}$ satisfying $|a_{\mathcal{S}_\tau}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$, the solution $\mathbf{w}^*$ defined in Eq. (3.17) achieves the global minimum of $J(\mathbf{w})$.*

Proof of Theorem 3.4.7. If $\tau = d$, we have $|w_{\mathcal{S}_i}^*| > 0$ for $i = 1, \cdots, d$, and each $w_{\mathcal{S}_i}^*$ satisfies the optimality condition given in Eq. (3.14). Thus, $\mathbf{w}^*$ is the global minimizer of $J(\mathbf{w})$.

If $\tau < d$, we have $|w_{\mathcal{S}_i}^*| > 0$ for $i = 1, \cdots, \tau$, and $|w_{\mathcal{S}_i}^*| = 0$ for $i = \tau + 1, \cdots, d$. Since τ is the largest coordinate of $\mathcal{S}$ satisfying $|a_{\mathcal{S}_\tau}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$, for $(\tau+1)$-th coordinate of $\mathcal{S}$, we have $|a_{\mathcal{S}_{\tau+1}}| - \frac{\lambda(\tau+1)}{1+\lambda(\tau+1)}\mu_{\tau+1} < 0$, which can be rewritten equivalently as $|a_{\mathcal{S}_{\tau+1}}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau < 0$, i.e., $|a_{\mathcal{S}_{\tau+1}}| < \lambda\|\mathbf{w}\|_1$. This implies $w_{\mathcal{S}_{\tau+1}}^*$ satisfies the optimal condition defined in Eq. (3.15). In a similar way, we can obtain that $w_{\mathcal{S}_i}^*$ satisfies the optimal condition defined in Eq. (3.15) for $i = \tau + 2, \cdots, d$.

On the other hand, $w_{\mathcal{S}_i}^*$ satisfies the optimal condition defined in Eq. (3.14) for $i = 1, \cdots, \tau$, according to Eq. (3.20).

Thus, $\mathbf{w}^*$ is the global minimizer of $J(\mathbf{w})$, which completes the proof. □

Once the largest τ is found out, we can easily obtain the optimal solution $\mathbf{w}^*$ via Eq. (3.17). Here is the question: how to find the largest coordinate τ of $\mathcal{S}$ satisfying $|a_{\mathcal{S}_\tau}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$?

In the following, we introduce an efficient algorithm to search the largest coordinate τ satisfying $|a_{\mathcal{S}_\tau}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$ in linear time, given d-dimensional vector $\mathcal{S}$ representing indexes of the descending order $|a_{\mathcal{S}_1}| \geq |a_{\mathcal{S}_2}| \geq \cdots \geq |a_{\mathcal{S}_d}|$.

Algorithm 2 Search the largest coordinate τ of $\mathcal{S}$.

Input: $\mathbf{a} \in \mathbb{R}^d$, $\mathcal{S} \in \mathbb{R}^d$, λ.

Output: τ, μ_τ.

```
Initialize: $\tau = d$, $\mu_\tau = \frac{1}{d}\sum_{i=1}^{d} |a_i|$.
while $\tau > 1$ do
  if $|a_{\mathcal{S}_\tau}| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$ then
    break.
  else
    $\mu_\tau = \frac{\tau}{\tau-1}\mu_\tau - \frac{1}{\tau-1}|a_{\mathcal{S}_\tau}|$.
  end if
  Set $\tau = \tau - 1$.
end while
return $\tau$, $\mu_\tau$.
```

3.4.3 The Summary of Exclusive Sparsity

In Theorem 3.4.5 and Algorithm 2, an sorting-based explicit approach is proposed to solve the $\ell_{1,2}$-norm based optimization problem (3.10). Based on abovementioned results, we explain the exclusive sparsity of $\ell_{1,2}$-norm in two aspects:

- ***Competitive:*** *in the competition, winner i in $\mathbf{w}$ represents $|w_i^*| > 0$, if $|a_i| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau > 0$; while, loser j in $\mathbf{w}$ represents $|w_j^*| = 0$, if $|a_j| - \frac{\lambda\tau}{1+\lambda\tau}\mu_\tau < 0$.*

- ***Survival**: if $\|\mathbf{w}\|_1 = 0$, then we have $|a_i| = 0$ for $i = 1, \cdots, d$, according to the optimality condition in Eq. (3.15). Thus, given $\mathbf{a} \neq \mathbf{0}$, the optimal solution $\mathbf{w}^*$ cannot become to a zero vector. That is to say, at least one element w_i in $\mathbf{w}$ can survive with $|w_i| > 0$.*

3.5 Optimization Algorithm

For purpose of selecting robust and flexible features, we are interested in the following optimization problem

$$\min_{\mathbf{W}} \; J_{\text{ex21}}(\mathbf{W}) = \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{W}\|_{1,2}^2 \tag{3.22}$$

where the least square loss is penalized by the proposed "exclusive $\ell_{2,1}$" regularization, and α, β are hyperparameters.

First, we add an auxiliary variable $\mathbf{Z}$ to make the optimization separable between $\ell_{2,1}$-norm and $\ell_{1,2}$-norm. Thus, original problem (3.22) becomes

$$\begin{aligned} \min_{\mathbf{W},\mathbf{Z}} \quad & \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{Z}\|_{1,2}^2 \\ \text{s.t.} \quad & \mathbf{Z} = \mathbf{W}. \end{aligned} \tag{3.23}$$

Then, augmented Lagrangian multiplier (ALM) [43] method is applied to enforce the constraint of problem (3.23) explicitly

$$\min_{\mathbf{W},\mathbf{Z}} \; \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{Z}\|_{1,2}^2 + \langle \mathbf{\Lambda}, \mathbf{Z} - \mathbf{W} \rangle + \frac{\nu}{2}\|\mathbf{Z} - \mathbf{W}\|_F^2 \tag{3.24}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, i.e. $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{ij} A_{ij}B_{ij}$, $\mathbf{\Lambda}$ is the Lagrangian multiplier, and ν is the penalty parameter.

Problem (3.24) can be further rewritten as

$$\min_{\mathbf{W},\mathbf{Z}} \; \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{Z}\|_{1,2}^2 + \frac{\nu}{2}\|\mathbf{Z} - \mathbf{W} + \mathbf{\Lambda}/\nu\|_F^2 . \tag{3.25}$$

Thus, our tasks are solving the variables $\mathbf{Z}$, $\mathbf{W}$ and updating the parameters $\mathbf{\Lambda}$, ν.

3.5.1 Solving for $\mathbf{Z}$

Firstly, we solve $\mathbf{Z}$ while fixing $\mathbf{W}$. As a result, problem (3.25) w.r.t $\mathbf{Z}$ becomes

$$\mathbf{Z}_{t+1} = \arg\min_{\mathbf{Z}} \ \frac{\nu_t}{2} \|\mathbf{Z} - \mathbf{W}_t + \boldsymbol{\Lambda}_t/\nu_t\|_F^2 + \beta\|\mathbf{Z}\|_{1,2}^2 \,. \tag{3.26}$$

Since the optimizations w.r.t each row of $\mathbf{Z}$ are separable, we can minimize problem (3.26) in a row-wise fashion. Thus, the optimization in Eq. (3.26) w.r.t $\mathbf{z}^i$ becomes

$$\mathbf{z}_{t+1}^i = \arg\min_{\mathbf{z}^i} \ \frac{\nu_t}{2} \left\|\mathbf{z}^i - \mathbf{e}\right\|_2^2 + \beta\left\|\mathbf{z}^i\right\|_1^2 , \tag{3.27}$$

where $i = 1, \cdots, d$ is the feature/row index, $\mathbf{e} = \mathbf{w}_t^i - \boldsymbol{\lambda}_t^i/\nu_t$, $\mathbf{w}_t^i$ is the i-th row of $\mathbf{W}_t$, and $\boldsymbol{\lambda}_t^i$ is the i-th row of $\boldsymbol{\Lambda}_t$.

Using Theorem 3.4.5, the optimal solution of Eq. (3.27) is given by

$$\mathbf{z}_{t+1}^i = \mathrm{sign}(\mathbf{e}) \odot \left[|\mathbf{e}| - \frac{2\beta\tau}{\nu_t + 2\beta\tau} \mu_\tau \right]_+ \tag{3.28}$$

where τ, μ_τ are computed using Algorithm 2, given the input ($\mathbf{e}$, $2\beta/\nu_t$, $\mathcal{S}$), and $\mathcal{S}$ is a k-dimensional vector representing the indexes of descending order with respect to $\mathbf{e}$, i.e., $|e_{\mathcal{S}_1}| \geq |e_{\mathcal{S}_2}| \geq \cdots \geq |e_{\mathcal{S}_k}|$.

3.5.2 Solving for $\mathbf{W}$

Secondly, we solve $\mathbf{W}$ while fixing $\mathbf{Z}$. As a result, problem (3.25) w.r.t $\mathbf{W}$ becomes

$$\mathbf{W}_{t+1} = \arg\min_{\mathbf{W}} \ \left\|\mathbf{X}^T\mathbf{W} - \mathbf{Y}\right\|_F^2 + \alpha\|\mathbf{W}\|_{2,1} + \frac{\nu_t}{2}\|\mathbf{Z}_{t+1} - \mathbf{W} + \boldsymbol{\Lambda}_t/\nu_t\|_F^2 \,. \tag{3.29}$$

Since $\ell_{2,1}$-norm is defined on each row $\mathbf{w}^i$ in $\mathbf{W}$, here we also can solve $\mathbf{W}$ in the similar way as $\mathbf{Z}$.

To solve $\mathbf{W}$ in a row-wise fashion, we decompose the least square loss w.r.t $\mathbf{w}^i$ as follows

$$
\begin{aligned}
\left\|\mathbf{X}^T\mathbf{W}-\mathbf{Y}\right\|_F^2 &= \left\|\sum_{i=1}^{d}(\mathbf{x}^i)^T\mathbf{w}^i-\mathbf{Y}\right\|_F^2 \\
&= \left\|(\mathbf{x}^i)^T\mathbf{w}^i-\mathbf{Y}^{-i}\right\|_F^2 \qquad (3.30)\\
&= a\|\mathbf{w}^i\|_2^2-2\mathbf{w}^i\mathbf{b}^T+c
\end{aligned}
$$

where $\mathbf{Y}^{-i}=\mathbf{Y}-\sum_{j\neq i}(\mathbf{x}^j)^T\mathbf{w}^j$, $a=\|\mathbf{x}^i\|_2^2$, $\mathbf{b}=\mathbf{x}^i\mathbf{Y}^{-i}$, and $c=\mathrm{Tr}((\mathbf{Y}^{-i})^T\mathbf{Y}^{-i})$.

Thus, the optimization in Eq. (3.29) w.r.t $\mathbf{w}^i$ becomes

$$
\mathbf{w}_{t+1}^i=\arg\min_{\mathbf{w}^i}\ a\|\mathbf{w}^i-\mathbf{b}/a\|_2^2+\alpha\left\|\mathbf{w}^i\right\|_2+\frac{\nu_t}{2}\left\|\mathbf{z}_{t+1}^i-\mathbf{w}^i+\boldsymbol{\lambda}_t^i/\nu_t\right\|_F^2. \qquad (3.31)
$$

where $i=1,\cdots,d$ is the feature/row index, $\mathbf{z}_{t+1}^i$ is the i-th row of $\mathbf{Z}_{t+1}$, and $\boldsymbol{\lambda}_t^i$ is the i-th row of $\boldsymbol{\Lambda}_t$.

Eq. (3.31) can be further rewritten as

$$
\mathbf{w}_{t+1}^i=\arg\min_{\mathbf{w}^i}\ \frac{2a+\nu_t}{2}\left\|\mathbf{w}^i-\mathbf{d}\right\|_2^2+\alpha\left\|\mathbf{w}^i\right\|_2 \qquad (3.32)
$$

where $\mathbf{d}=\frac{1}{2a+\nu_t}(2\mathbf{b}+\nu_t\mathbf{z}_{t+1}^i+\boldsymbol{\lambda}_t^i)$.

Theorem 3.5.1. *The optimal solution of*

$$
\min_{\mathbf{w}}\ \|\mathbf{w}-\mathbf{a}\|_2^2+\lambda\|\mathbf{w}\|_2
$$

is given by

$$
\mathbf{w}^*=\max(1-\frac{\lambda}{2\|\mathbf{a}\|_2},0)\mathbf{a}.
$$

Proof of Theorem 3.5.1. Since $\mathbf{w}=\rho\mathbf{a}$ ($\rho\in\mathbb{R}^+$), the objective function becomes

$$
\lambda\rho\|\mathbf{a}\|_2+(\rho-1)^2\|\mathbf{a}\|_2^2.
$$

By setting the derivative of the objective function w.r.t ρ to zero, we have

$$
\rho^*=\max(1-\frac{\lambda}{2\|\mathbf{a}\|_2},0).
$$

With the relation that $\mathbf{w}^* = \rho^* \mathbf{a}$, the optimal solution of $\mathbf{w}$ is obtained as

$$\mathbf{w}^* = \max(1 - \frac{\lambda}{2\|\mathbf{a}\|_2}, 0)\mathbf{a},$$

which completes the proof. □

Using Theorem 3.5.1, the optimal solution of Eq. (3.32) is given by

$$\mathbf{w}_{t+1}^i = \left[1 - \frac{\alpha}{(2a+\nu)\|\mathbf{d}\|_2}\right]_+ \mathbf{d}, \tag{3.33}$$

where $[x]_+ = \max(x, 0)$.

3.5.3 Updating Parameters

Finally, we update parameters $\mathbf{\Lambda}$, ν at the end of t-th iteration as the following

$$\mathbf{\Lambda}_{t+1} = \mathbf{\Lambda}_t + \nu_t(\mathbf{Z}_{t+1} - \mathbf{W}_{t+1}), \tag{3.34}$$

$$\nu_{t+1} = \rho\nu_t \tag{3.35}$$

where $\rho > 1$ is a constant.

3.5.4 The Summary of Optimization Algorithm

The complete framework of the proposed augmented Lagrangian multiplier (ALM) based optimization algorithm is summarized in Algorithm 3.

3.6 Experiments

3.6.1 Benchmark Datasets

Extensive experiments on twelve benchmark datasets are performed to evaluate the effectiveness of feature selection methods on multi-class classification problems. Among those benchmarks, there are 4 image datasets: MNIST[2] [34], Yale[3], YaleB[4],

[2] In MNIST, 100 images are randomly selected out of each digit.

[3] `http://vision.ucsd.edu/content/yale-face-database`

[4] `http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html`

Algorithm 3 ALM based optimization algorithm for solving the "exclusive $\ell_{2,1}$" regularization based problem (3.22).

```
Input: data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, class labels $\mathbf{Y} \in \mathbb{R}^{n \times k}$, hyperparameters $\alpha$, $\beta$.
Output: weight matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$.
Initialize: $t = 0$, $\nu_t = 1/\|\mathbf{X}\|_F$, $\rho = 1.1$, $\epsilon_1 = 1e{-}8$, $\epsilon_2 = 1e{-}5$, $\mathbf{\Lambda}_t = \mathbf{0}$, random
    initialization weights $\mathbf{W}_t$.
repeat
    for $i \in \{1, \cdots, d\}$ do
        Compute $\mathbf{e}$ via Eq. (3.27).
        Compute the descending order $\mathcal{S}$ of $\mathbf{e}$.
        Compute $\tau$, $\mu_\tau$ via Algorithm 2 given ($\mathbf{e}$, $2\beta/\nu_t$, $\mathcal{S}$).
        Compute $\mathbf{z}^i_{t+1}$ via Eq. (3.28).
    end for
    for $i \in \{1, \cdots, d\}$ do
        Compute $\mathbf{Y}^{-i}$, $a$, $\mathbf{b}$ via Eq. (3.30).
        Compute $\mathbf{d}$ via Eq. (3.32).
        Compute $\mathbf{w}^i_{t+1}$ via Eq. (3.33).
    end for
    Update $\mathbf{\Lambda}_{t+1}$ via Eq. (3.34).
    Update $\nu_{t+1}$ via Eq. (3.35).
    Set $t = t + 1$.
until Convergence condition is satisfied:
    $|J_{\text{ex21}}(\mathbf{W}^{t+1}) - J_{\text{ex21}}(\mathbf{W}^t)|/J_{\text{ex21}}(\mathbf{W}^t) \leq \epsilon_1$,
    $\|\mathbf{Z}^{t+1} - \mathbf{W}^{t+1}\|_\infty \leq \epsilon_2$.
return The optimal solution of weight matrix: $\mathbf{W}^*$.
```

PIE [44]; 1 spoken letter recognition dataset: ISOLET[5]; 5 bio-microarray datasets: Carcinomas [36], Lung [37], Glioma [45], TOX [38], Tumor-14 [46]; and 2 text datasets: CNAE-9 [47], 20-Newsgroups[6]. The detail of twelve benchmark datasets is summarized in Table 3.4.

Dataset	k	n	d
MNIST	10	1000	784
Yale	15	165	1024
YaleB	38	2414	1024
PIE	10	210	2420
ISOLET	26	1560	617
Carcinomas	11	174	9182
Lung	5	203	3312
Glioma	4	50	4434
TOX	4	171	5748
Tumor-14	14	190	16063
CNAE-9	9	1080	856
20-Newsgroups	20	2000	5000

Table 3.4: The summary description of twelve benchmark datasets. k, n, d denote the number of classes, the number of data instances, the number of features for each dataset, respectively.

3.6.2 Evaluation Metrics

In the experiments, our proposed exclusive $\ell_{2,1}$ regularization based feature selection method is compared to five state-of-the-arts, including three filter methods: *F-statistic* [10], *ReliefF* [11], *minimum redundancy maximum relevance (mRMR)* [12], and two sparse coding based methods: *multi-task feature selection via $\ell_{2,1}$-norm* ($\ell_{2,1}$) [16, 17, 18, 19], *exclusive Lasso* (*eLASSO*) [26, 28].

[5] `http://featureselection.asu.edu/datasets.php`

[6] `http://qwone.com/~jason/20Newsgroups/` In 20-Newsgroups, 100 documents are randomly selected out of each newsgroup, and F-statistic method is used to prescreen 5,000 keywords.

Towards evaluating the performance on classification, five-fold cross-validation accuracy with SVM classifier are computed on average for each feature selection method. We use LIBSVM [39] as the practical implementation of SVM, where the kernel is set as linear and the parameter C is set as 1 (the default value) for all the experiments.

When training different sparse coding based models, hyperparameters are adjusted to enforce the same level of sparsity in the learned weight matrix $\mathbf{W}$ for each method.

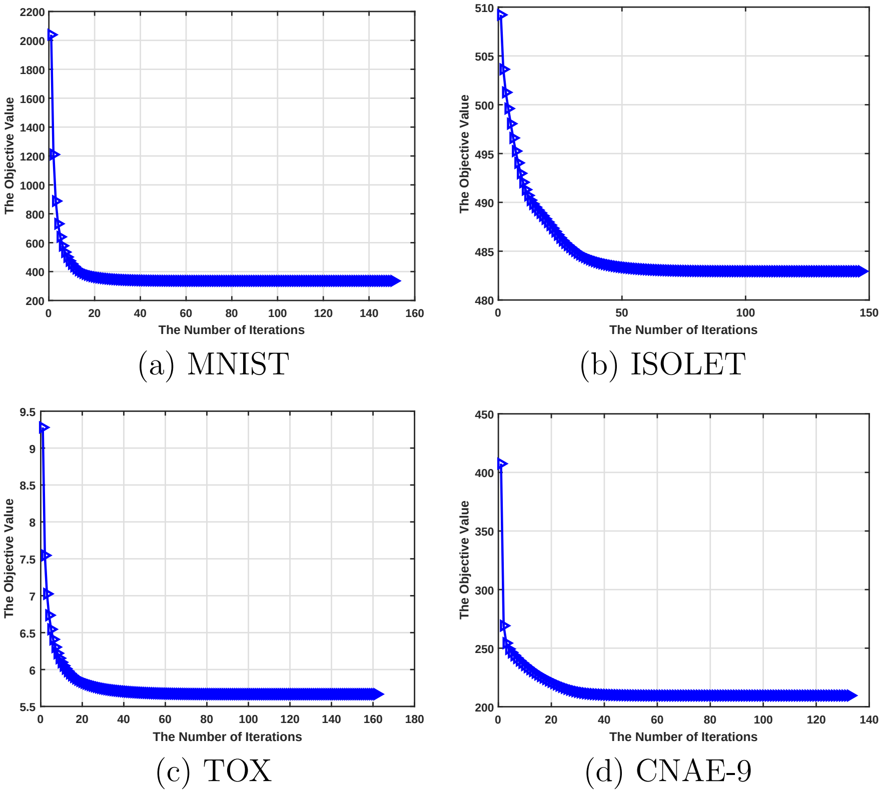
For testing, three filter methods and $\ell_{2,1}$ build k SVM classifiers, and each SVM classifier uses the same feature subset, since they select a subset of features shared by all the classes jointly. On the contrary, eLASSO and the proposed "exclusive $\ell_{2,1}$" method build k SVM classifiers, and each SVM classifier uses different feature subsets, since they select discriminative features for each class separately. Then, the final classification result is obtained via majority voting.

3.6.3 Analysis of the Results

3.6.3.1 Convergence Study

The convergence of our proposed ALM based optimization algorithm is shown in Figure 3.1, where x-axis and y-axis denote the number of iterations and the objective value respectively.

We used the same hyperparameter setting ($\alpha = 1$, $\beta = 1$) for four benchmark datasets. As it can be seen in Figure 3.1, our proposed optimization algorithm takes around 100$\sim$150 iterations to converge. Even though it is difficult to optimize two non-smooth terms (i.e. $\ell_{2,1}$-norm and $\ell_{1,2}$-norm) simultaneously, our ALM based algorithm is very efficient, and can converge fast in real applications.



(a) MNIST

(b) ISOLET

(c) TOX

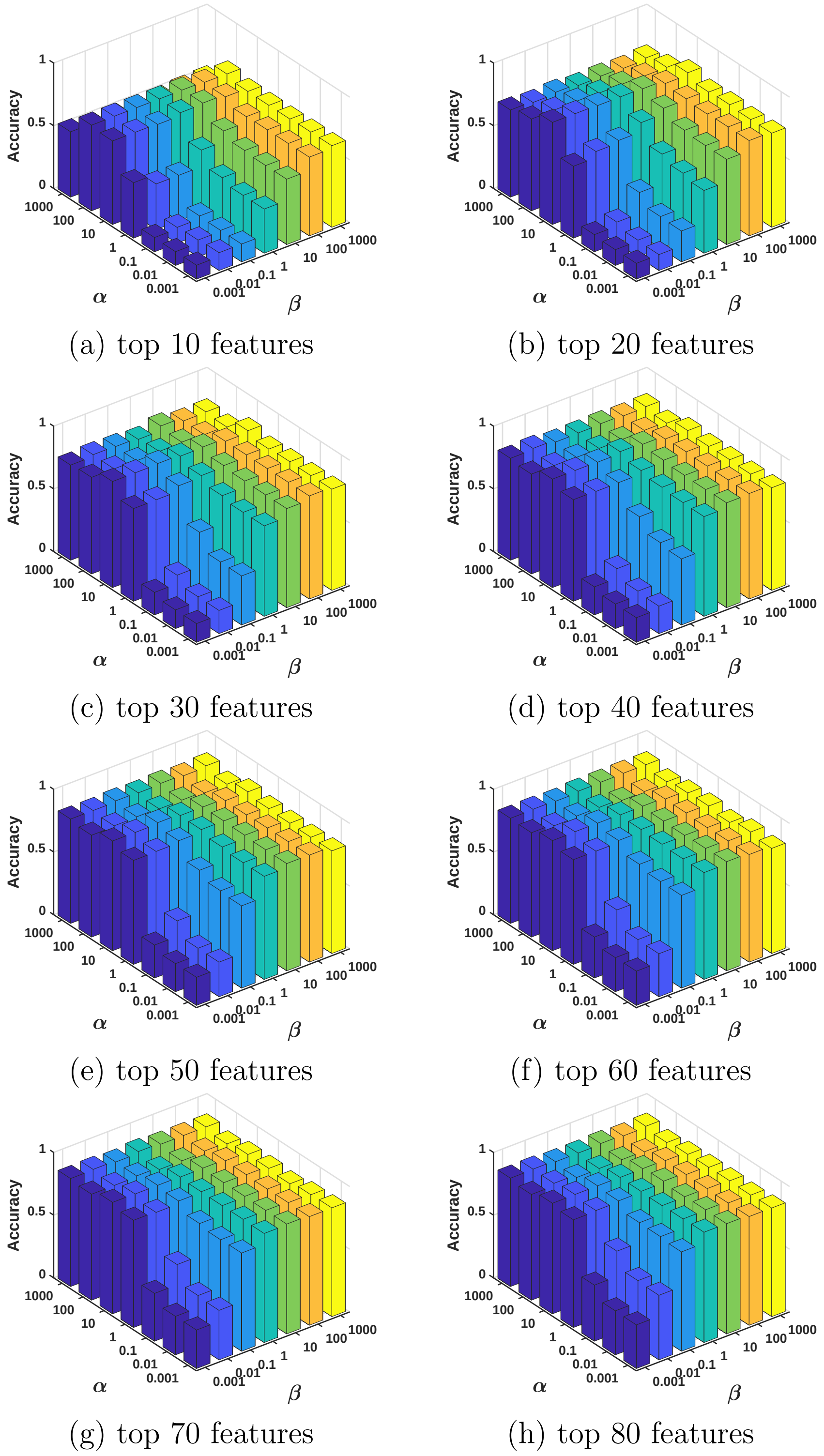
(d) CNAE-9

Figure 3.1: Convergence analysis of the proposed optimization algorithm on four benchmark datasets.

3.6.3.2 Parameter sensitivity study

The effect of hyperparameters on the performance of our proposed feature selection method is studied in this section. The relation between α and β on MNIST and TOX dataset is shown in Figures 3.2-3.3, where x-axis (the value of α) and y-axis (the value of β) are changing from 0.001 to 1000, and z-axis is the five-fold cross-validation classification accuracy. Besides, we increase the number of selected features from (a) to (h), using top 10∼80 features. With the increasing of α and β, we have more zero values (less features) in weight matrix $\mathbf{W}$.

The classification performance is relatively lower when α and β are very small or large, since small regularization strength is unable to find the structural sparsity in



(a) top 10 features

(b) top 20 features

(c) top 30 features

(d) top 40 features

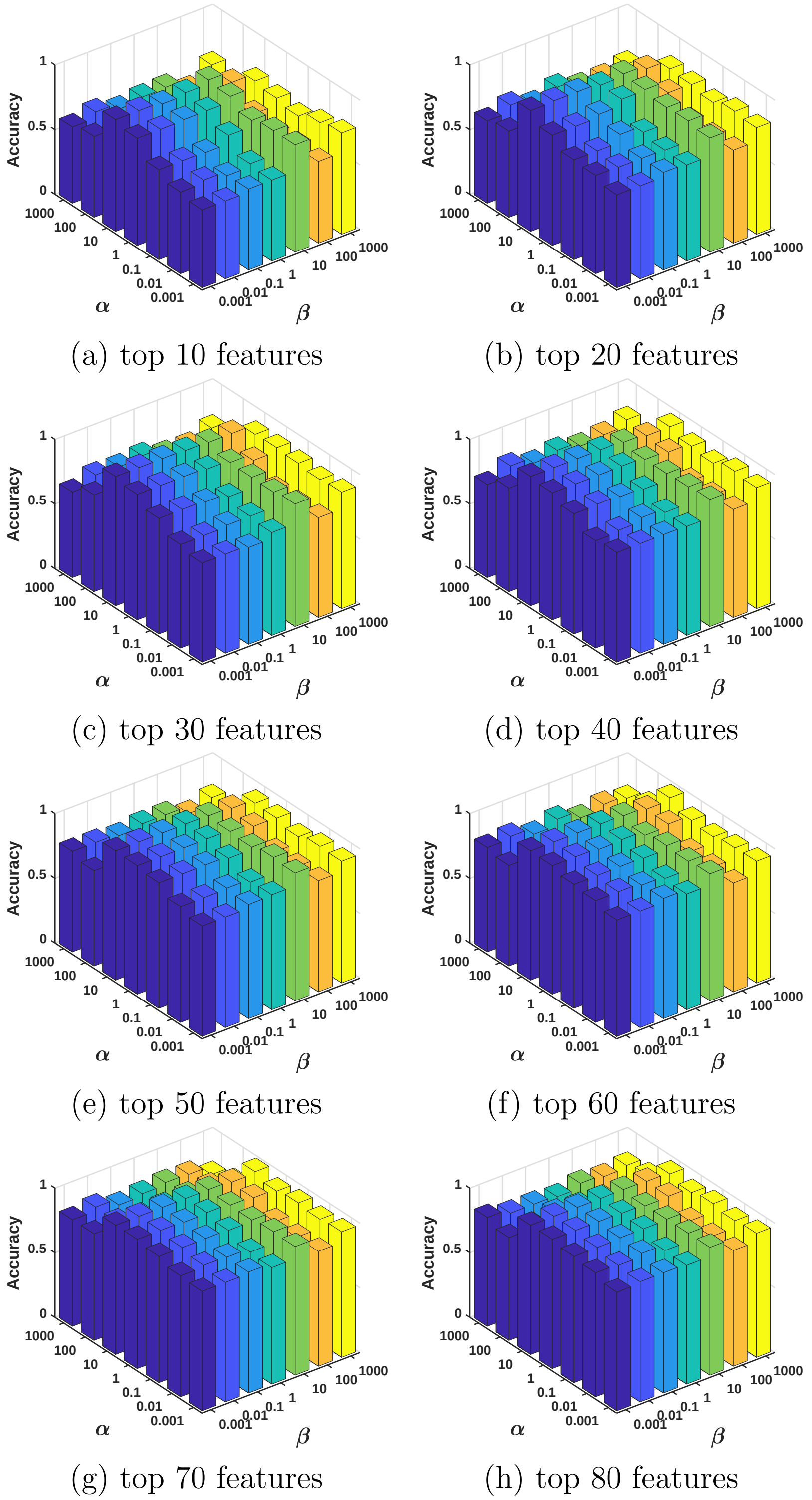
(e) top 50 features

(f) top 60 features

(g) top 70 features

(h) top 80 features

Figure 3.2: Parameter sensitivity analysis on MNIST dataset.



(a) top 10 features

(b) top 20 features

(c) top 30 features

(d) top 40 features

(e) top 50 features

(f) top 60 features

(g) top 70 features

(h) top 80 features

Figure 3.3: Parameter sensitivity analysis on TOX dataset.

$\mathbf{W}$, and large regularization strength penalizes the loss severely thus cannot effectively preserve the relation between $\mathbf{X}$ and $\mathbf{Y}$.

Overall, our method is not sensitive to the hyperparameter, and does not change too much while varying the value of α and β in different feature settings.

3.6.3.3 Classification Results Comparison

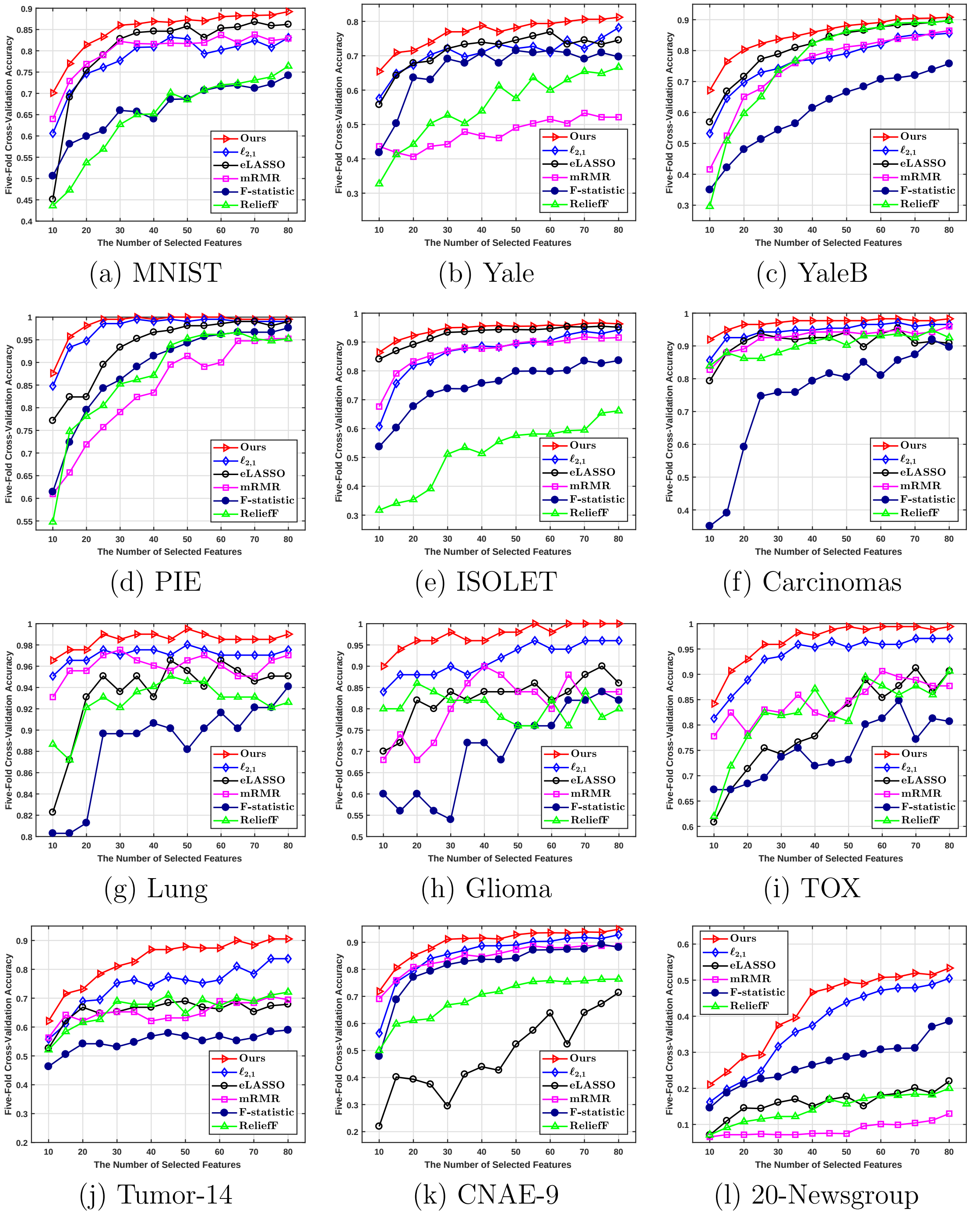
Experimental results of our proposed robust flexible feature selection method versus five state-of-the-arts on twelve benchmark datasets are shown in Figure 3.4, where x-axis denotes the number of selected features ranging from 10 to 80 with the interval equal to 5, and y-axis denotes the average of five-fold cross-validation classification accuracy.

In general, sparse coding based methods ($\ell_{2,1}$, *eLASSO*, *Ours*) achieve better performances than filter methods (*F-Statistic*, *ReliefF*, *mRMR*). Among those filter methods, *mRMR* has relatively higher classification accuracy in most cases, since it takes consideration of minimizing the correlation between features.

eLASSO performs better than filter methods in image and spoken letter recognition datasets. However, its performance has a great degradation in bio-microarray and text datasets, since $\ell_{1,2}$-norm cannot remove a large amount of irrelevant noise features in high-dimensional data space.

$\ell_{2,1}$ has a very stable performance in all the datasets via selecting class-shared features. In some cases, $\ell_{2,1}$ performs even close to *our method* around top 60$\sim$80 features.

Overall, *our method* obtains the best classification result on twelve benchmark datasets. Additionally, in the small number of selected features setting, such as top 10$\sim$20, *our method* has an overwhelming advantage over other methods, with around 5%$\sim$10% improvement on accuracy.



(a) MNIST (b) Yale (c) YaleB

(d) PIE (e) ISOLET (f) Carcinomas

(g) Lung (h) Glioma (i) TOX

(j) Tumor-14 (k) CNAE-9 (l) 20-Newsgroup

Figure 3.4: Five-fold cross-validation accuracy of the proposed feature selection method versus state-of-the-arts on twelve benchmark datasets.

3.7 Conclusion

In this work, we introduce a novel exclusive $\ell_{2,1}$ regularization for robust flexible feature selection. Besides, a sorting-based explicit approach is proposed to solve $\ell_{1,2}$-norm, which further explains the exclusive sparsity of $\ell_{1,2}$-norm. Then, an efficient augmented Lagrangian multiplier based optimization algorithm is proposed to iteratively solve the exclusive $\ell_{2,1}$ regularization in a row-wise fashion. Experimental results on twelve benchmark datasets verify the effectiveness of our proposed robust flexible feature selection method, which outperforms five state-of-the-art methods on multi-class classification problems.

CHAPTER 4

DEEP ROBUST DATA RECONSTRUCTION USING L1-AUTOENCODER NETWORKS

4.1 Introduction

Data reconstruction/recovery is the important topic in machine learning and computer vision areas. The task aims at automatically recovering the original clean data from the corrupted input in unsupervised manner, also known as denoising [6] and noise reduction [7] in other domains.

Traditional machine learning methods, such as principal component analysis (PCA) [48], treat images as one-dimensional vector. This data format is convenient for subspace learning. Given the input data with corruption, PCA tries to approximately reconstruct the original data in a low-dimensional space, via minimizing the squared error of the difference between corrupted input and reconstruction. However, outliers usually dominate the objective function of PCA due to the enlarged residuals after squaring, which leads to the instability and ineffectiveness of reconstruction. For purpose of diminishing this adverse influence, L_1-PCA [49] is proposed to reform PCA into the absolute value of the residuals using L_1 loss. Since dealing with each residual equally, L_1-PCA is less sensitive to outliers, and even capable of generating more robust and stable reconstructed results. On the other hand, Zhang et al of [50] applies augmented Lagrange multipliers (ALM) [51] to minimize the non-differentiable L_1-norm, which decomposes the original problem into a sequence of sub-problems with closed-form solutions. In addition, robust PCA (RPCA) [52] presents the original PCA using a new formulation that the corrupted input data is equivalent to the summation of the reconstruction and the additive errors. To resolve this special constraint, Wright et al of [52] propose proximal gradient (PG) with continuation [53] to optimize a surrogate loss, which enforces nuclear norm [54] on reconstruction and L_1-norm on additive errors, so as to achieve a sparse and low-rank solutions. Thus it can be seen that traditional methods focus on obtaining a low-rank approximation of the original data. Nevertheless, they only take consideration of shallow models using a simple and straightforward representation of the input data.

Recently, deep neural network is very popular in many research areas by means of non-linear activations and multi-layer representations. As compared to traditional machine learning methods, performance on a variety of problems in many fields have been greatly improved through deep learning, such as classification [55], clustering [56], object detection [57], semantic segmentation [58], speech recognition [59], image caption [60], etc.

Among those applications using deep network, autoencoder (AE) [61] is most relevant to data reconstruction topic, because its optimal solution when using linear activation is strongly related to PCA. Autoencoder learns a nonlinear low-dimensional latent representation for a set of data. To prevent overfitting, autoencoder enforces regularization on parameters of deep network by weight decay [62] and dropout [63]. For data reconstruction, squared error (L_2 loss) and cross-entropy loss are widely used to define the network's reconstruction loss.

Additionally, denoising autoencoder (dAE) [64] has been proposed to achieve robust reconstruction. However, it is different from unsupervised methods that the locations of corruption in the images are commonly available to perform the pretraining for dAE. In addition, the Gaussian noise assumption makes dAE deal with only small noises. In practice, larger noises are usually existing in real applications, which are most suitably treated with L_1 loss.

Although L_1 loss has been mentioned in several deep learning works, such as [65, 66, 67, 68, 69], to the best of our knowledge, there is no existing systematic investigation on deep network with L_1 loss, and no work on deep learning using L_1 for data reconstruction in unsupervised manner.

Motivated by recent works, we propose a deep network in the form of autoencoder with L_1 loss to achieve robust reconstruction and improve performance on corrupted input data. As compared to traditional machine learning methods, our model learns a deep and non-linear representation of the input data, and obtain much better reconstruction quality against a variety of image occlusions.

A key contribution here is that we resolve the black spots problem in robust data recovery using ReLU as activation with L_1 loss. ReLU (rectified linear unit) is widely adopted in deep learning due to its fast convergence [70], and good performance on many tasks. However, our experiments in autoencoder networks for data recovery reveal a new and interesting fact:

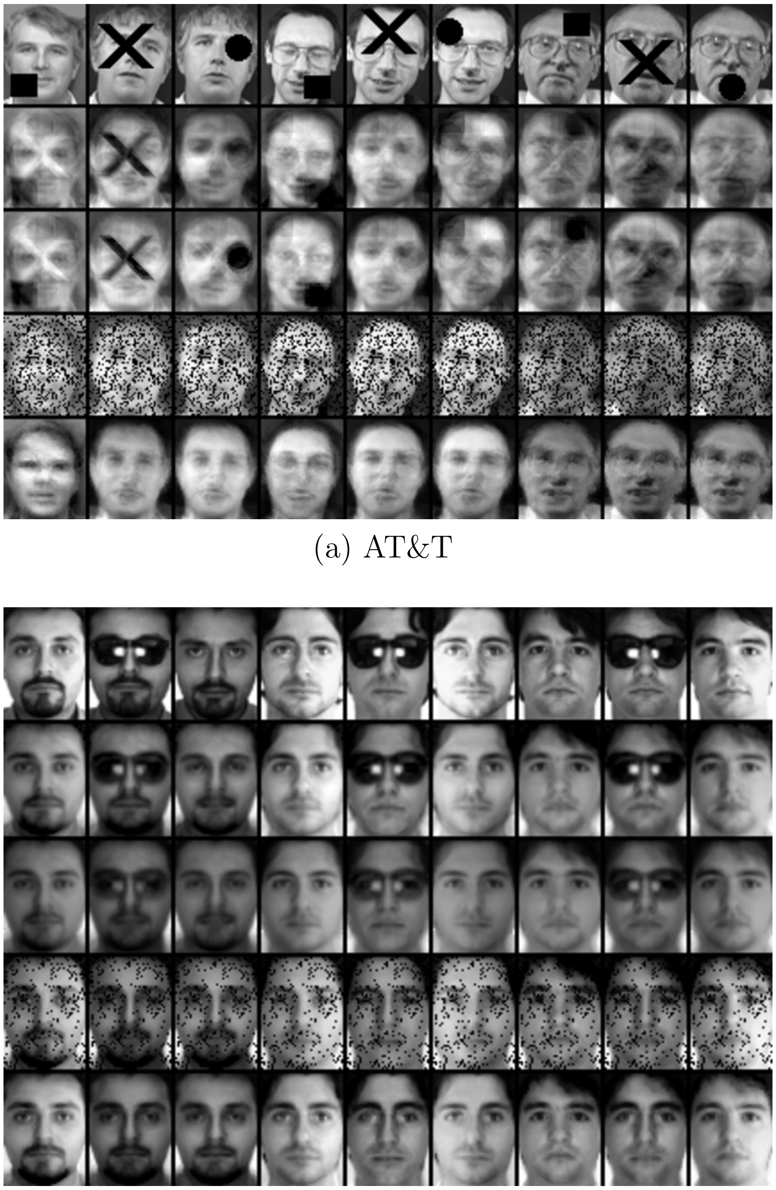
- *Autoencoder using ReLU with cross-entropy loss or L_2 loss produces outputs (which is the desired reconstructed images) without black spots.*
- *Autoencoder using ReLU with L_1 loss produces output with black spots.*

The experiment results are shown in Fig. 4.1. This new finding presents a challenge in robust data reconstruction/recovery, since L_1 loss can do the robust recovery while cross-entropy or L_2 loss can not.

We analyzed ReLU activation function and find a smoothed version of ReLU working with L_1 loss can effectively remove black spots. This works for all the datasets we have done in the experiments (see Fig. 4.10).

Our experimental results (see Fig. 4.7) also show that increasing the number of layers in multi-layer L_1-autoencoder framework with smoothed ReLU as activation steadily improves the quality of reconstruction (i.e. the output of the network).

Thus, the deep autoencoder network is a good network structure that provides a deep representations of the input, and effectively eliminates occlusions as inconsistent with main intrinsic properties of a set of data.



(a) AT&T

(b) AR

Figure 4.1: Black spot problem. Top: AT&T faces. Bottom: AR faces. 1st row: input corrupted images. 2nd-5th rows: output of 3-layer autoencoder using L_2+ReLU (2nd), cross-entropy+ReLU (3rd), L_1+ReLU (4th), L_1+sReLU (5th).

4.2 Deep Robust Data Reconstruction

A set of input data is represented by matrix $X \in R^{p\times n}$, where $X = (x_1, \cdots, x_n)$. A data can be one-dimensional vector or two-dimensional matrix, such as a r-by-c image I represented as $\mathrm{vec}(I) \in R^p$, where $p = r \cdot c$ is the total number of pixels. The code $Z = (z_1, \cdots, z_n) \in R^{d\times n}$ is the lower-dimensional ($d \ll p$) latent representation of original input data. The reconstructed output is $\widehat{X} = (\widehat{x}_1, \cdots, \widehat{x}_n) \in R^{p\times n}$, recovered from Z.

In real life applications, the input data can be noisy, corrupted, etc. Our task is to reconstruct the noise-free data $\widehat{X}$ from the corrupted input X automatically. The difficulty is that the recovery is done in an unsupervised manner, i.e., no prior knowledge on the noise. For example, the locations of those occlusions are unknown (see Fig. 4.1).

4.2.1 Robust Autoencoder with ℓ_1 Loss

In this work, we propose a robust L_1-autoencoder network for data reconstruction. The $(2\ell+1)$-layers autoencoder (AE) [61] network has total 2ℓ nested functions, where the code is generated as

$$Z = f_\ell\{\cdots f_3\{f_2[f_1(X, \theta_1), \theta_2], \theta_3\} \cdots, \theta_\ell\}, \tag{4.1}$$

and the output of the entire network is generated as

$$\widehat{X} = f_{2\ell}\{\cdots f_{\ell+3}\{f_{\ell+2}\{f_{\ell+1}(Z, \theta_{\ell+1}), \theta_{\ell+2}\}, \theta_{\ell+3}\} \cdots, \theta_{2\ell}\}, \tag{4.2}$$

where θ_i is the parameters in i-th layer.

Suppose the output of $(i-1)$-th layer is H_{i-1}, then we have $H_i = f_i(H_{i-1}, \theta_i) = \sigma(\theta_i H_{i-1})$, where $\sigma(\cdot)$ is the activation function, such as tanh, sigmoid, and ReLU (rectified linear unit) [55].

For robust reconstruction, we are interested in the following L_1 loss based optimization problem

$$\min_{\theta_1,\cdots,\theta_{2\ell}} \left\| X - \widehat{X} \right\|_1 + \lambda \left(\sum_{i=1}^{2\ell} \frac{1}{2} \|\theta_i\|_2^2 \right), \tag{4.3}$$

where λ is the hyperparameter controlling the importance of the L_2-regularization term.

A multi-layer ℓ_1-autoencoder network is shown in Fig. 4.2. The top subfigure is the process of initializing network's parameters using stacked autoencoder (SAE) [64] method, where the hidden output of the i-th autoencoder is feeded to $(i+1)$-th autoencoder as the input. Then, we repeat this process until all the l autoencoders are initialized. Finally, those l encoders are unrolled with tied weights and concatenated to form a deep L_1-autoencoder network for robust data reconstruction (see the bottom subfigure).

Thus, we have $\theta_i = \{W_i, b_i\}$, for $i = 1, \cdots, l$, and $\theta_i = \{W_{2l+1-i}^T, b_{2l+1-i}^T\}$, for $i = l+1, \cdots, 2l+1$. Here, we use the ℓ_1 loss to measure the absolute difference between the corrupted input data and the reconstructed output.

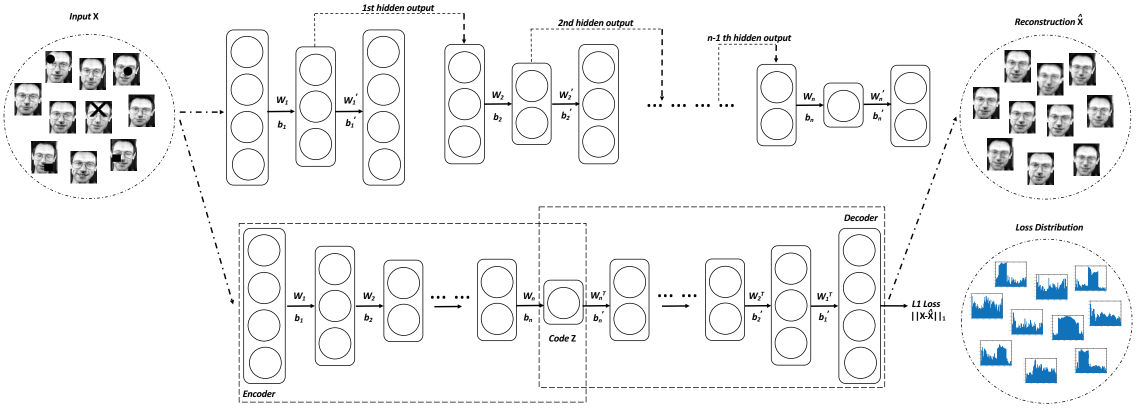


Figure 4.2: Deep robust reconstruction network (bottom) and the parameter initialization (top).

4.2.2 Analysis of Black Spot Problem

Compared to tanh and sigmoid activations, rectified linear unit (ReLU) activation [55] can substantially accelerate the convergence of deep network using gradient descent, due to its linear and non-saturating form. However, in autoencoder networks using L_1 loss, we find the output images have many pixels (neurons) with zero values, which gives the black spots in the greyscale image as shown in Fig. 4.1.

Leaky ReLU is an available choice to overcome this problem. However its output has negative values while the input image data are nonnegative. Thus, leaky ReLU is not suitable.

Our finding (as explained in section 4.1 Introduction) that L_1 loss with ReLU outputs the image with black spot while cross-entropy or L_2 loss with ReLU do not have black spot problem motivates us to consider the computational aspects of these loss functions, when the network output value $\hat{x}$ is small and positive, since this is the region that the black spot problem occurs.

A complete analysis of the nonlinear network is difficult. Here we do analysis on highly simplified network structure with network internal parameters ignored. Let x be the network input. L_1 loss is $\ell_1 = |\hat{x} - x|$, L_2 loss is $\ell_2 = (\hat{x} - x)^2$, cross entropy loss is $\ell_{CE} = -x \log \hat{x} - (1 - x) \log(1 - \hat{x})$.

Clearly, L_1 loss has large (magnitude 1) positive or negative gradient: $\partial \ell_1 / \partial \hat{x} = \text{sign}(\hat{x} - x)$. Cross-entropy loss has only large negative gradient $\partial \ell_{CE} / \partial \hat{x} = -x/\hat{x} + (1 - x)/(1 - \hat{x})$. L_2 loss has small gradient $\partial \ell_2 / \partial \hat{x} = 2(\hat{x} - x)$ (in L_2 loss, $x \simeq \hat{x}$).

All networks use gradient descent type algorithm. Thus when $\hat{x}$ is small, L_1 loss could drive $\hat{x}$ even smaller or zero when the gradient is positive, and cause the black spot problem. Cross entropy will drive $\hat{x}$ larger by the large negative gradient. L_2 loss will not change $\hat{x}$ much. Thus cross-entropy and L_2 loss unlikely cause the black spot problem.

4.2.3 Smoothed ReLU (sReLU) Activation

The gradient of the loss function affects the output layer through the gradient of activation function. Thus, we study the gradient of activation function.

The ReLU activation $\sigma(z) = \max(z, 0)$ has the gradient $\sigma'(z) = 1$ for $z > 0$ and $\sigma'(z) = 0$ for $z \leq 0$. This activation gradient behavior combined with the gradient from the L_1 loss causes the black spot problem.

The reasoning is the following. (i) When $\hat{x}$ is small and positive. This implies the activation has an input z from the hidden layers which is small and positive. Thus $\sigma'(z) = 1$ is large. This combined with large positive gradient from the L_1 loss helps drive $\hat{x}$ to zero. (ii) When $\hat{x} = 0$. This implies $z \leq 0$. Thus $\sigma'(z) = 0$. There is little chance that $\hat{x}$ will rise from zero. These two facts cause the black spot problem.

This analysis motivates us to propose a smoothed version of ReLU such that when $z \leq 0$, the activation has a small nonzero gradient. With this change, when $\hat{x} = 0$, the small gradient from the activation combined with the large negative gradient of the L_1 loss (x is always nonnegative) will help $\hat{x}$ to rise up from zero.

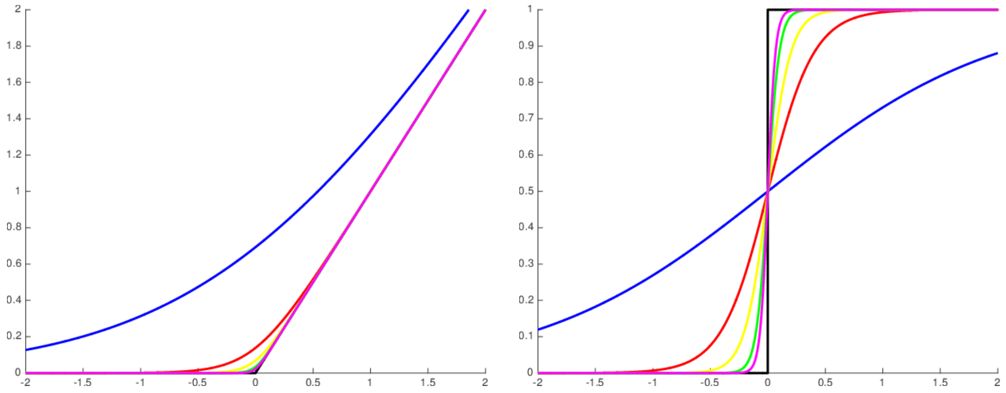
The smoothed ReLU is motivated in the following way. First, the ReLU function $\max(x, 0)$ can be expressed as

$$\max(x, 0) = \lim_{s \to \infty} \frac{1}{s} \log(1 + e^{sx}). \tag{4.4}$$

Thus, we define the smoothed ReLU (sReLU) activation function $\sigma(x)$ as

$$\sigma(x) = \frac{1}{s} \log(1 + e^{sx}). \tag{4.5}$$

The sReLU function and its gradients are shown in Fig. 4.3. sReLU is much more smooth than ReLU, especially the gradients. The case when $s = 1$ is same as the softplus function [71]. However, as shown in Fig. 4.3, $s = 1$ deviates significantly from the ReLU and is not a good approximation. In practical applications, we use



(a) the function itself

(b) gradient of the function

Figure 4.3: ReLU versus sReLU. Left: the function itself. Right: gradient of the function. Black is ReLU. Blue, red, yellow, green and megenta are sReLU with s=1, 5, 10, 20 and 30 respectively.

$s = 10 \sim 30$. The gradient of sReLU is $\sigma'_{\text{sReLU}}(x) = 1/(1 + e^{-sx})$, which is almost identical to the classical sigmoid function $\sigma_{\text{sigmoid}}(x) = 1/(1 + e^{-x})$. This intrinsic connection to the sigmoid function remains to be further explored.

Reconstructed result using ReLU and sReLU as activation is shown in Fig. 4.1. Black spots appear in ReLU based networks with L_1 loss. More details are explained in section 4.2.4.

4.2.4 Network Structure

4.2.4.1 Mathematical Formulation of Autoencoder

The simplest autoencoder for reconstruction with L_1 loss is a three layer neural network. With input X, hidden Z, and reconstruction $\widehat{X}$ defined as

$$A_1 = W_1 X + b_1, \tag{4.6}$$

$$Z = f_{\theta_1}(X) = \sigma(A_1), \tag{4.7}$$

$$A_2 = W_2 Z + b_2, \tag{4.8}$$

$$\widehat{X} = f_{\theta_2}(Z) = \sigma(A_2), \tag{4.9}$$

where $\theta_1 = \{W_1, b_1\}$, $\theta_2 = \{W_2, b_2\}$, and $\sigma(\cdot)$ is sReLU.

In the following experiments, we also construct 5-layer and 7-layer networks. Results on benchmark datasets show deeper networks steadily improve the quality of reconstructed results. The mathematical formulation of multi-layer autoencoders is very similar to above formulae for 3-layer autoencoder, and will not be repeated here.

4.2.4.2 Memory-efficient Gradient of sReLU

We strive to save memory in actual computer implementation. When using sigmoid activation, because of the relationship $\sigma'_{\text{sigmoid}}(x) = \sigma_{\text{sigmoid}}(x)(1 - \sigma_{\text{sigmoid}}(x))$, the derivatives are not computed and not saved in memory.

For network using ReLU activation, this technique can not be utilized. But for network using sReLU, this technique can be used, because the gradient of sReLU activation can be expressed as

$$\sigma'_{\text{sReLU}}(x) = 1 - e^{-s\sigma_{\text{sReLU}}(x)}, \tag{4.10}$$

Thus the gradient need not to be computed and stored.

4.2.4.3 Approximation to L_1 Loss

These autoencoders use smoothed ReLU as activation and L_1 loss of Eq.(4.3). The L_1 loss $\|A\|_1 = \sum_{ij} |A_{ij}|$ involves function $f(x) = |x|$ which is non-differentiable at $x = 0$, and has large gradient jump from $f'(x) = -1$ to $f'(x) = 1$ as x chances small negative value to a small positive value. To remove the non-differentiability and numerical uncertainty when x is close to zero, we use a smoothed version to approximate the L_1 loss term

$$\left\| X - \widehat{X} \right\|_1 \simeq \sqrt{\left\| X - \widehat{X} \right\|_F^2 + \epsilon^2}, \tag{4.11}$$

where ϵ is set to a small value depending on application. For image recovery, the input X and output $\widehat{X}$ are pixel values. When image pixel values are on 0 to 255 scale, setting $\epsilon = \sqrt{0.1}$ is sufficient since the minimum pixel value difference is 1. If we normalize each pixel value to be within $[0, 1]$, then setting $\epsilon = \sqrt{0.1}/256$ is sufficient. Compared to hard L_1 loss, when x changes from $x = -\epsilon$ to $x = \epsilon$, the gradient of L_1 loss changes from -1 to +1, but the gradient of smoothed L_1 loss changes very little. Thus the smoothed L_1 loss significantly improves the numerical stability.

4.2.5 Network Parameter Initialization

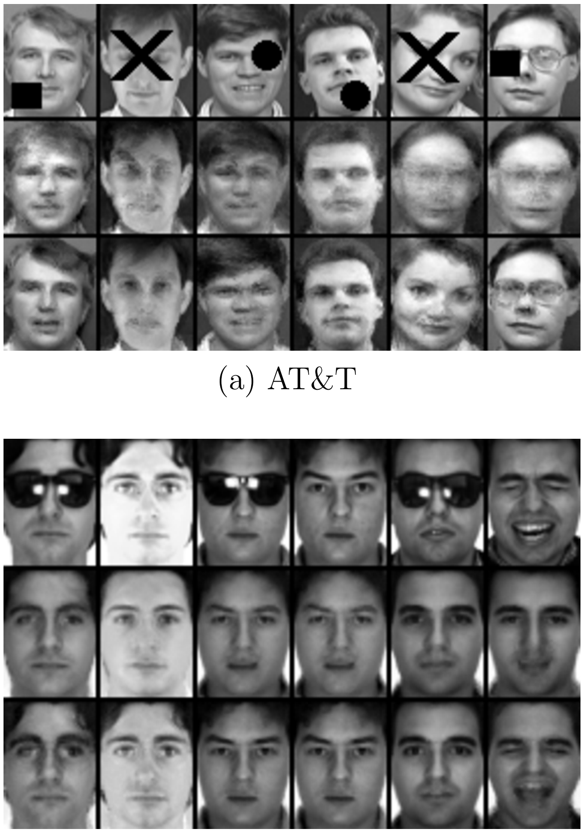
Traditionally autoencoder are pretrained by restricted Boltzmann machine [72], which is most suitable, however, for sigmoid activation based model. This pretraining strategy is inappropriate to initialize parameters of the proposed autoencoder with L_1 loss especially using smoothed ReLU as activation. Otherwise, finetuning stage will converge early and get stuck at a bad local minima with high loss, due to unmatched activation units. Consequently, it always generates poorly reconstructed results.

The efficient pretraining on the proposed L_1 loss autoencoder using sReLU proceeds this way. We initialize parameters in stepwise fashion using stacked autoencoder (SAE) [73] approach, where the input of $(i+1)$-th encoder is assigned as the hidden unit output of the i-th encoder. Connection weights in single autoencoder are generated using Xavier's initializer [74]. After greedy pretraining, all the encoders are unrolled with tied weights and concatenated to form multi-layer autoencoder, which then is finetuned to minimize the L_1 loss for robust data reconstruction.

4.2.6 Network Output Layer: No Activation

The output layer of most autoencoders use same activations as inner layers, for example [61]. This is natural. The input image pixel values are nonnegative;

the output before activation (A_2 in Eq.(4.9)) has mixed signs. Thus the activation naturally bring values to nonnegative, comparable to input X as in Eq.(4.3). Thus, $\widehat{X} \geq 0$ can be satisfied automatically after activation.



(a) AT&T

(b) AR

Figure 4.4: ReLU versus sReLU. Top: AT&T faces; Bottom: AR faces. 1st row: input corrupted images. 2nd-3rd rows: output of 7-layer autoencoder using L_1+ReLU (2nd), L_1+sReLU (3rd).

For having better understanding of whether the above black spot problem is due to ReLU (in context of L_1 loss), we do experiment on autoencoder network where no ReLU activation is applied at output layer. The results are shown in Fig. 4.4. Even though black spot disappeared in the network without ReLU activation, the quality of the output is not as good as the output of sReLU based networks.

4.3 Experiments and Analysis

4.3.1 Benchmark Datasets

To evaluate the proposed robust data reconstruction model, extensive experiments on two benchmark datasets (i.e. AT&T and AR) are performed to recover the original clean data from the corrupted input data in this section.

AT&T: faces are collected by AT&T Laboratories Cambridge, including ten different images of each of 40 distinct subjects. Under varying lighting and facial expression, frontally upright images were taken at different times.

AR: cropped images [8] from 100 individuals are obtained in two sections. Each person has 14 natural face images (different expression and illumination) and 12 occluded face images (sun glasses and scarf).

The detail of benchmark datasets is summarized in Table 4.1, where face images are transformed to fixed ratios in greyscale mode. In AT&T dataset, 40 persons are equally divided into four groups. Similarly, 20 men and 20 women from AR dataset are selected out of 100 individuals to form four groups.

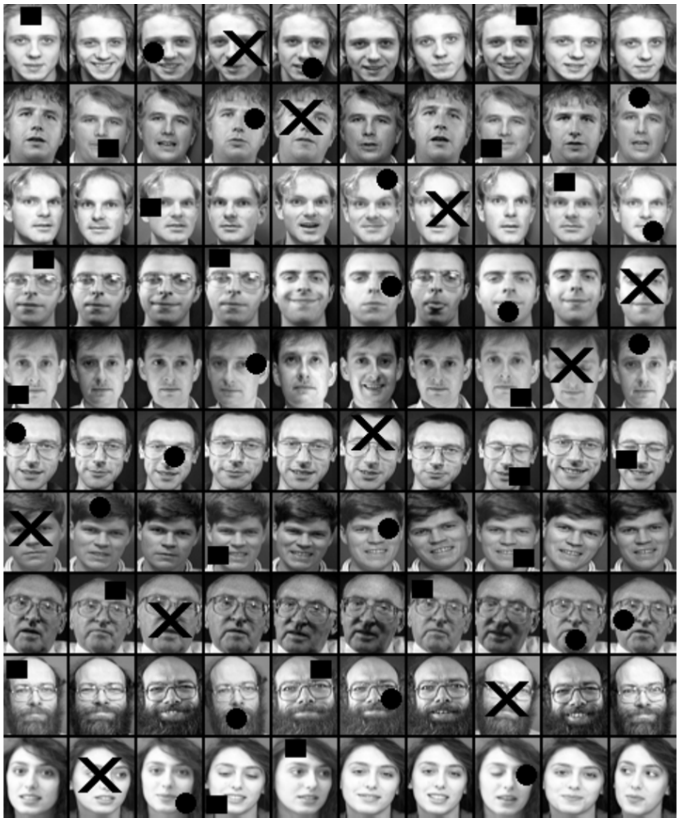
Dataset	#Images	#Dimensions	#Class
AT&T	400	$56\times46=2576$	40
AR	2600	$55\times40=2200$	100

Table 4.1: Description of benchmark datasets.

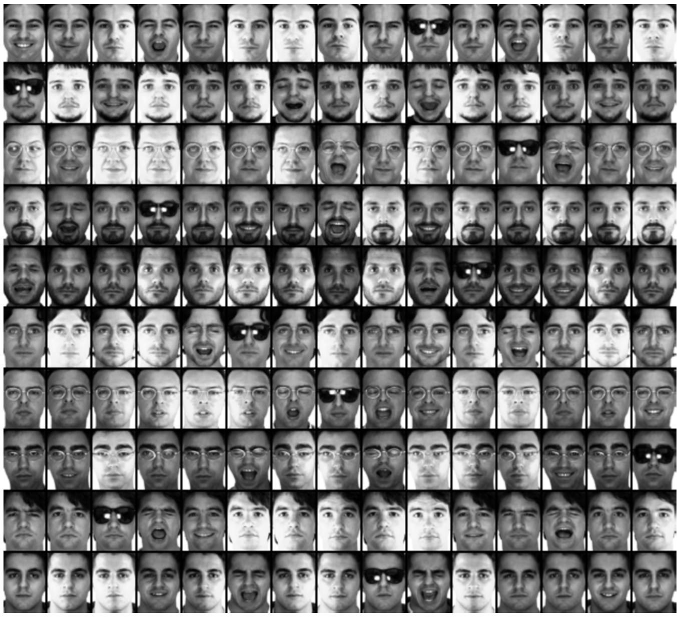
4.3.2 Occluded Images

For purpose of studying the effectiveness of reconstruction methods, in subsequent experiments, four occlusions are added into images to evaluate the performance qualitatively and quantitatively.

Details of the occlusions are shown in Fig. 4.5.



(a) AT&T



(b) AR

Figure 4.5: Occluded images. Top: AT&T faces; Bottom: AR faces.

In AT&T dataset, three types of synthetic occlusions are randomly added into ten face images of each individual, including one cross, two rectangles, and two circles (see Fig. 4.5-(a)). Pixels in the occlusion are set to zero values (black pixels).

In AR dataset, face images are naturally occluded by sun glasses and scarf. Since the scarf has large size, it is difficult to reconstruct original faces without heavy distortion. Thus, for each individual, only one sun glasses image (see Fig. 4.5-(b)) is selected as real occlusion to form the input with other fourteen natural faces.

4.3.3 Evaluation metrics

Standard unsupervised metric and protocols are used to study the effectiveness of reconstruction. We evaluate the performance using relative noise-free reconstruction error (ERR) defined as

$$ERR = \frac{\left\| X_0 - \widehat{X} \right\|_F}{\left\| X_0 \right\|_F}, \tag{4.12}$$

where X_0 is the original uncorrupted image (Ground Truth), and $\widehat{X}$ is the result reconstructed from corrupted image $X = X_0 + E$. In unsupervised learning tasks, E represents unknown different type of occlusions.

Since ground truth is not available for sun glasses occlusion in AR, reconstruction error is calculated with respect to other fourteen uncorrupted images, only if the occlusion is removed successfully. Additionally, errors of four different groups are marked as 1st-4th columns respectively in Figs. 4.6, 4.8, 4.9, and the average of four errors is shown in the legend for each methods.

4.3.4 Network Implementation

The performance of our L_1 loss network with sReLU as activation is investigated in different layers/depths.

The relevant setting of the architecture of L_1-autoencoder networks is given in Table 4.2.

Data set	3 Layers (3L)	5 Layers (5L)	7 Layers (7L)
AT&T	2576-8-2576	2576-20-8-20-2576	2576-30-20-8-20-30-2576
AR	2200-9-2200	2200-20-9-20-2200	2200-30-20-9-20-30-2200

Table 4.2: The architecture of L_1-autoencoder networks for AT&T and AR.

In training, each batch we feed face images of ten persons from one group (AT&T: 100 images, AR: 150 images) into the network. Hyperparameter λ is searched in the set $\{0.0001,\ 0.001,\ 0.01, 0.1\}$ to achieve the best result. Scale parameter s in smoothed ReLU is fixed as 10.

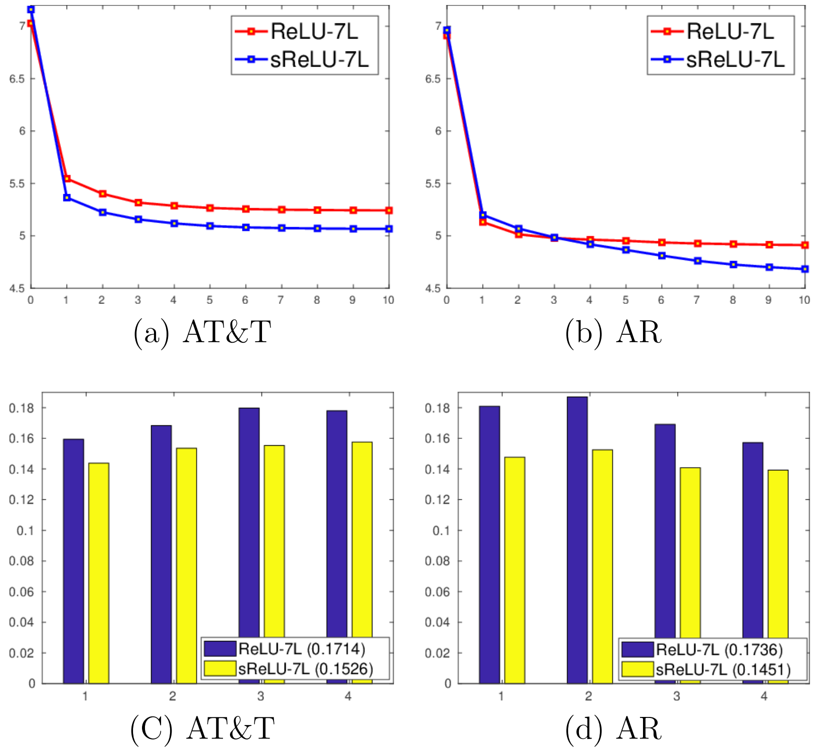
Towards fast convergence, conjugate gradient (CG) is used to pretrain parameters of each autoencoder in 200 epochs. Then, the unrolled multi-layer autoencoder is finetuned by limited-memory BFGS (L-BFGS) [75] in 10 epochs, where the max iteration of L-BFGS optimizer is set to 400.

4.3.5 Deep ℓ_1-Autoencoder Result: ReLU versus sReLU

Here we present the proposed L_1-autoencoder results, emphasizing the comparison between sReLU and ReLU.

As it can be seen in Fig. 4.1 that even though occlusions are removed, there are many black spots in reconstructed faces by L_1+ReLU. Conversely, black spots disappeared in the output of other networks. Nevertheless, L_2+ReLU and cross-entropy+ReLU fail to remove most of the occlusions.

On the contrary, L_1+sReLU achieves the best result that all the occlusions are eliminated, and the network output does no produce any black spots. This is the motivation for smoothed ReLU (sReLU) to replace hard ReLU.



(a) AT&T

(b) AR

(C) AT&T

(d) AR

Figure 4.6: Comparison of L_1+ReLU versus L_1+sReLU. (a,b): Convergence of log of training objectives. (c,d): Noise-free reconstruction error.

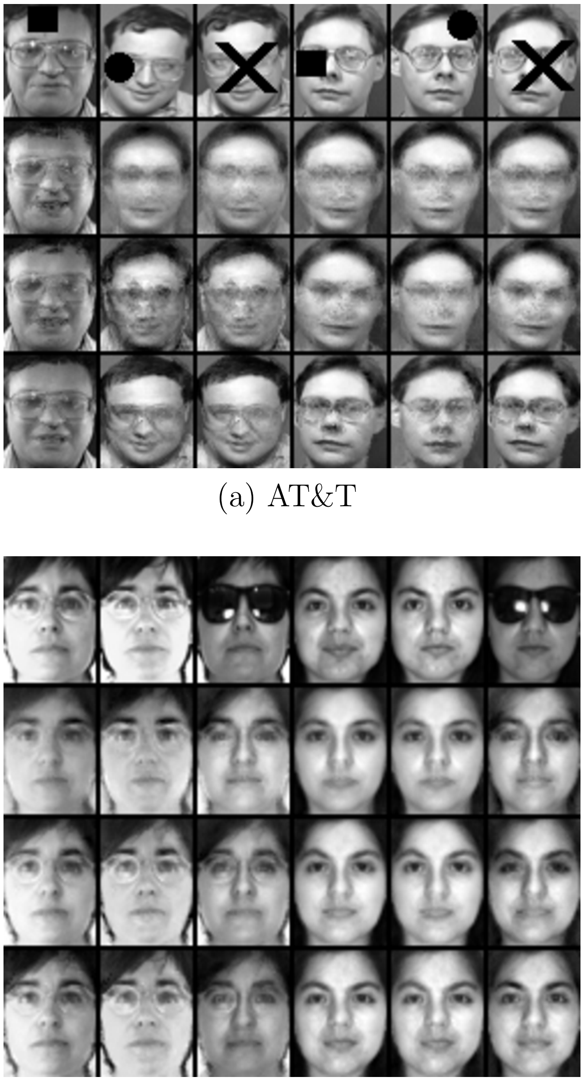
On the other hand, we did the experiment without activation at the output layer of the network, mentioned in sec 4.2.6. It is obvious in Fig. 4.4 that the output of L_1-autoencoder using sReLU recovers more facial details, which outperforms the reconstruction results using ReLU.

Due to numerical instability incurred by non-smooth gradients, networks using ReLU have early convergence problem, see Fig 4.6-(a,b). That is the reason why sReLU based networks achieve lower noise-free reconstruction error (ERR) than ReLU based networks, see Fig 4.6-(c,d).

Thus it can be seen that sReLU resolves black spot problem of ReLU as activation in L_1 loss networks. From visualized and statistical results, sReLU based network performs better in reconstruction than ReLU based network.

4.3.6 Deep versus Shallow ℓ_1-Autoencoder Networks

For purpose of further improving reconstructed results of 3-layer L_1-networks (see Fig. 4.1), more hidden layers are added into L_1-networks to build a deep robust reconstruction model. The comparison between shallow and deep L_1 networks are shown in Fig. 4.7.



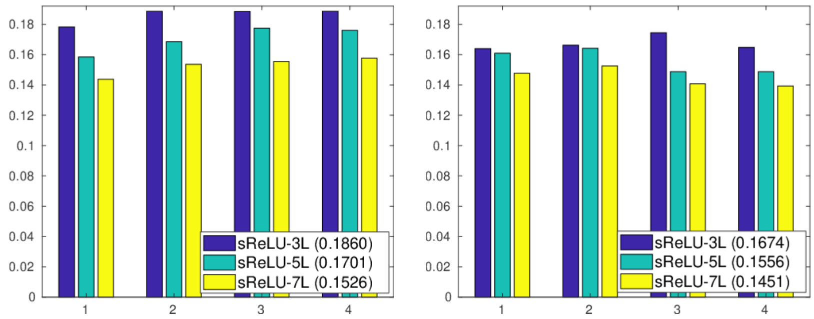
(a) AT&T

(b) AR

Figure 4.7: Deep versus shallow networks. Top: AT&T faces. Bottom: AR faces. 1st row: input corrupted images. 2nd row: output of 3-layer network. 3rd row: output of 5-layer network. 4th row: output of 7-layer network.

As it can be seen in Fig. 4.7, the quality of the reconstruction is improved steadily with the increase of the number of layers. Deeper 5-layer and 7-layer networks recover more facial details than 3-layer networks, such as hair and glass in AT&T faces, and eye and complexion in AR faces. Contrarily, shallow 3-layer network cannot capture the complex structure of the input data, so as to lose too much facial information.

Statistics in Fig. 4.8 also shows similar results that shallow L_1 network performs worse than deep L_1 network since 5/7-layer networks obtain lower noise-free reconstruction error.



(a) AT&T (b) AR

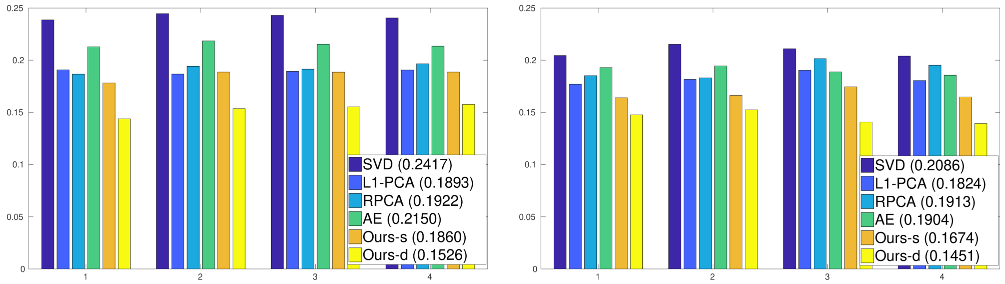
Figure 4.8: Deep versus shallow networks in relative noise-free reconstruction error (ERR). Left: AT&T faces. Right: AR faces.

4.3.7 Deep ℓ_1-Autoencoder versus State-of-the-arts

The proposed L_1-autoencoder using sReLU as activation is compared against state-of-the-arts, including traditional machine learning methods: singular value decomposition (SVD), L_1 loss based PCA (L_1-PCA) [49], and Robust PCA ($RPCA$) [52]. Additionally, deep learning method is taken into consideration such as sigmoid based autoencoder [61] (AE) using L_2 loss, which uses the same network architecture (see Table 4.2) as our method.

The relative noise-free reconstruction errors (ERR) is calculated on reconstructed results from five unsupervised algorithms, and is shown in Fig. 4.9 where statistical values in legend is the average of errors from four columns, since each benchmark dataset has been divided into four groups (explained in section 4.3.1).

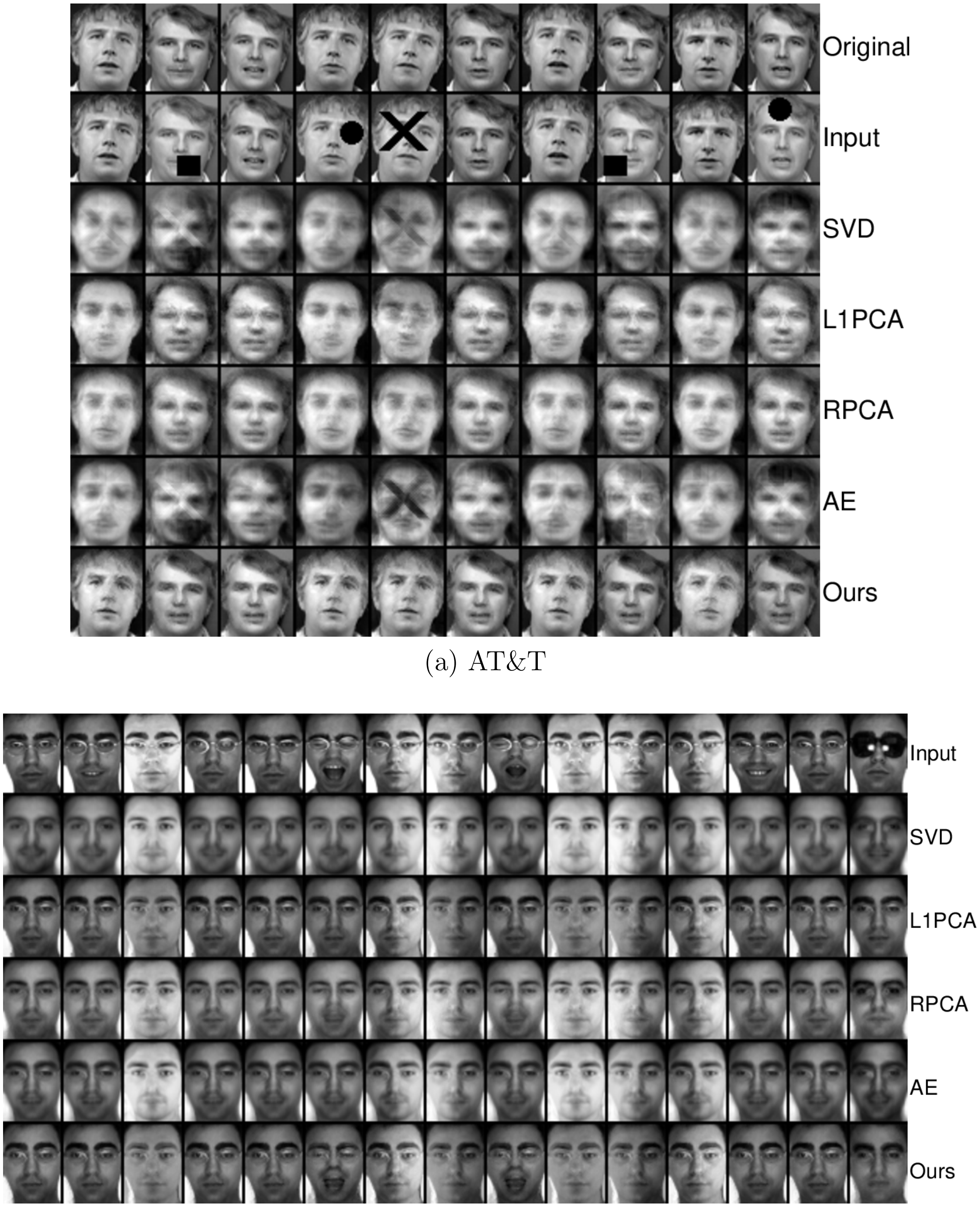
For traditional machine learning methods, model parameters of SVD, L_1-PCA, RPCA are finetuned to achieve the best robust and low-rank reconstruction results. For AE method, the best result obtained from the outputs of 3/5/7-layer networks is reported here.



(a) AT&T (b) AR

Figure 4.9: L_1-autoencoder versus state-of-the-arts in relative noise-free reconstruction error (ERR). Left: AT&T faces. Right: AR faces.

As shown in the Fig. 4.9, robust models (L_1-PCA, RPCA, Ours) has lower errors than L_2-loss model, such as SVD and AE. Because of non-linear transformations used in networks, AE obtains slightly better results as compared to SVD. Most importantly, the proposed L_1-autoencoder using sReLU have a statistically overwhelming advantage over other methods, with the lowest reconstruction errors. Additionally, performance can be further improved from shallow 3-layer network (Ours-s) to deep 7-layer network (Ours-d). Thus, our deep L_1 networks has the strongest robustness among those methods when dealing with various corrupted data. In summary, statistical results verify the effectiveness of our robust reconstruction method.



(a) AT&T

(b) AR

Figure 4.10: Deep L_1-autoencoder network versus state-of-the-arts. Top (a): AT&T faces. Bottom (b): AR faces. Results from five different methods.

In order to compare visualized performance among five unsupervised methods, reconstructed results from two persons in AT&T and AR are shown in Fig. 4.10. For each method, the best result by tuning parameters is shown in each row.

Firstly, the result on AT&T faces is analyzed as follows. Both SVD and AE fails to remove occlusions completely. Based on SVD's result, L_1-PCA improves performance a lot since it eliminates most of occlusions. RPCA has a slightly better reconstruction than L_1-PCA since it generates smoother results, for example 4th and 5th faces. Among these methods, our deep L_1-autoencoder achieves best reconstructed results, which substantially improves the facial details.

Secondly, it is the explanation for experimental results on AR faces. Towards removing sun glasses, SVD loses too much information so that recovered faces look totally different from original input faces. As compared to RPCA and AE, L_1-PCA completely eliminates all the dark pixels of sun glasses around the eye. L_1-autoencoder network also achieves best result that occlusions are removed and more facial details are recovered, such as facial expression around the mouth at 6th and 9th faces.

4.4 Conclusions

In this work, we presented deep autoencoder networks with L_1 loss for robust data reconstruction in presence of corrupted images. We found that naively using L_1 loss with popular ReLU often has black spot problem. Analysis of this problem leads us to introduce a smoothed ReLU activation that effectively resolves the black spot problem associated with ReLU as activation. The proposed reconstruction method is capable of removing various occlusions added into face images, meanwhile recovering as many originally uncorrupted pixels as possible without any distortions. Extensive experimental results show that our deep robust reconstruction method outperforms state-of-the-arts quantitatively and qualitatively.

CHAPTER 5

ROBUST PCA BASED LOW-RANK AND SPARSE DATA RECONSTRUCTION

5.1 Introduction

Nowadays, data usually lies in thousand- or even million-dimensional observation spaces, such as image, audio, video, web, bioinformatic, etc. Thus, finding the lower-dimensional structure of high dimensional data becomes an very important task in machine learning and data mining areas.

Principal component analysis (PCA) [48] is the widely used method for lower-dimensional subspace learning and dimension reduction. Lots of research works indicate that the data usually have a low intrinsic complexity, such as the data is low-rank [76], basis could be sparse [77], or the data lies in low-dimensional manifold [78, 79]. When we assume that data points lie in low-dimensional manifold and the manifold is linear or nearly-linear, the low-dimensional structure of data can be effectively captured by a linear subspace spanned by the principal PCA directions. PCA is exploiting the best low-rank representation of the given data, and the optimal solution of principal directions and principal components can be stably and efficiently computed by singular value decomposition (SVD).

In a d-dimensional space, n data points are represented as matrix form: $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$, where $\mathbf{x}_i \in \mathbb{R}^d$. Principal directions are defined as $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k) \in \mathbb{R}^{d \times k}$, and principal components are defined as $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_k)$ $\in \mathbb{R}^{n \times k}$, where $\mathbf{V}_{ij}$ means the i-th data's projection along the j-th principal direction. There are two formulations of standard PCA. One is the covariance based approach. First, we compute the covariance matrix $\mathbf{C} = \sum_{i=1}^{n} (\mathbf{x_i} - \overline{\mathbf{x}})(\mathbf{x_i} - \overline{\mathbf{x}})^T = \mathbf{X}\mathbf{X}^T$, where we assume the data are already centered, i.e. $\overline{\mathbf{x}} = \mathbf{0}$, and drop the constant factor $\frac{1}{n-1}$ which does not affect $\mathbf{U}$. Thus, the principal directions of PCA model are obtained as follow

$$\begin{aligned} \max_{\mathbf{U}} \quad & \mathrm{Tr}(\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U}\mathbf{U}^T = \mathbf{I}. \end{aligned} \tag{5.1}$$

Another one is the matrix low-rank approximation based PCA model, i.e., $\mathbf{X} \simeq \mathbf{U}\mathbf{V}^T$. The goal is to recover $\mathbf{U}$ and $\mathbf{V}$ via minimizing the squared error of the difference between the original data $\mathbf{X}$ and the low-rank approximation/reconstruction $\mathbf{U}\mathbf{V}^T$, which can be mathematically formulated as

$$\min_{\mathbf{U},\mathbf{V}} \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_F^2 = \sum_{ij} \left[\mathbf{X}_{ij} - \left(\mathbf{U}\mathbf{V}^T\right)_{ij}\right]^2, \tag{5.2}$$

where the rank of $\mathbf{U}$ and $\mathbf{V}$ is equal to k. In practical applications, $k \ll n$. The optimal solution of Eq. (5.2) can be efficiently obtained by singular value decomposition of the input data $\mathbf{X}$. According to SVD's result, we can further prove the equivalence between the solution of Eq. (5.1) and the solution of Eq. (5.2).

However, classical PCA using ℓ_2 loss becomes ineffective when dealing with grossly corrupted or outlying preservations [48]. Thus, classical PCA is not applicable to practical applications, since noises or outliers are usually existing in the collected input data. A number of approaches have been exploited to improve the robustness of PCA in the literature.

Towards achieving robust solutions in noisy environment, L_1-PCA [49] is proposed to minimize the following objective function, defined as the absolute value of the difference between the noisy input $\mathbf{X}_{ij}$ and the reconstruction $(\mathbf{U}\mathbf{V}^T)_{ij}$,

$$\min_{\mathbf{U},\mathbf{V}} \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 = \sum_{ij} \left|\mathbf{X}_{ij} - \left(\mathbf{U}\mathbf{V}^T\right)_{ij}\right|, \tag{5.3}$$

where ℓ_1-norm is defined as $||\mathbf{A}||_1 = \sum_{ij} |\mathbf{A}_{ij}|$. It is well known that ℓ_1-norm is much more robust to corruptions/outliers than ℓ_2-norm. The ℓ_1 loss models implicit noises of the input data using Laplacian distribution, as compared to Gaussian distribution assumed by ℓ_2 loss. Ke et al of [49] introduces an alternative optimization method to minimize the ℓ_1 loss. Unknown variables $\mathbf{U}$, $\mathbf{V}$ are alternatively minimized until the loss function converges. At each iteration, one unknown variable is solved by standard convex linear/quadratic programming, while fixing another unknown variable.

However, the computation cost of above-mentioned alternative algorithm is extremely expensive, since the optimization is conducted in column-wise fashion. As a result, Zhang et al of [50] proposes an efficient matrix based optimization algorithm. First, an auxiliary variable $\mathbf{E}$ is introduced into L_1-PCA, i.e., $\mathbf{E} = \mathbf{X} - \mathbf{U}\mathbf{V}^T$. Then, Eq. (5.3) can be rewritten equivalently as

$$\begin{aligned} \min_{\mathbf{E},\mathbf{U},\mathbf{V}} \quad & \|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} = 0. \end{aligned} \tag{5.4}$$

Then, the Lagrangian multiplier $\mathbf{A}$ and the penalty parameter μ are introduced to enforce the equality constraint explicitly. Thus, the augmented Lagrangian multiplier (ALM) based formulation of L_1-PCA is obtained as

$$\min_{\mathbf{E},\mathbf{U},\mathbf{V}} \|\mathbf{E}\|_1 + \langle \mathbf{A}, \mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} \rangle + \frac{\mu}{2} \|\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E}\|_F^2, \tag{5.5}$$

which can be decomposed into a sequence of sub-problems with respect to $\mathbf{E}, \mathbf{U}, \mathbf{V}$, and each sub-problem can be computed efficiently with closed-form solutions. As it can be seen that the computation is based on matrix manipulation, ALM based optimization algorithm is more efficient than alternative minimization using convex linear/quadratic programming.

Additionally, Robust PCA (RPCA) [52] reformulates classical PCA as $\mathbf{X} = \mathbf{A} + \mathbf{E}$, which aims at recovering a low-rank matrix $\mathbf{A}$ from corrupted input $\mathbf{X}$ with unknown sparse error $\mathbf{E}$. Because of the difficulty in directly minimizing the rank of

$\mathbf{A}$ and the sparsity (i.e. ℓ_0-norm) of $\mathbf{E}$, the following convex surrogate is introduced to achieve the low-rank reconstruction and the sparse error,

$$\begin{aligned} \min_{\mathbf{A},\mathbf{E}} \quad & \|\mathbf{A}\|_* + \lambda\|\mathbf{E}\|_1 \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{A} + \mathbf{E}. \end{aligned} \tag{5.6}$$

Then, proximal gradient (PG) [53] is applied to solve a relaxed version of problem (5.6), where the constraint is treated as a penalty term,

$$\min_{\mathbf{A},\mathbf{E}} \ \mu\|\mathbf{A}\|_* + \mu\lambda\|\mathbf{E}\|_1 + \frac{1}{2}\|\mathbf{X} - \mathbf{A} - \mathbf{E}\|_F^2. \tag{5.7}$$

The solution of sub-problems with respect to $\mathbf{A}, \mathbf{E}$ can be computed efficiently via soft-thresholding operator and singular value thresholding operator [54]. The optimization algorithm converges until the Frobenius norm of the subgradient of Eq. (5.7) becomes sufficiently small.

Inspired by previous works, we propose robust PCA based low-rank and sparse data reconstruction method. As compared to RPCA which models an additive error on the input data, our proposed robust method models additive errors on principal directions and principal components. Besides, additive errors are enforced to be sparse and bounded. Thus it can be seen our method models the noise of principal directions and principal components in lower-dimensional latent subspace, while RPCA method models the noise of the input data in original feature space.

However, the optimization becomes very difficult since four unknown variables are minimized simultaneously. At the same time, additive errors should be sparse and bounded. If we adopt alternative minimization strategy used in L_1-PCA [49], the optimization of the proposed loss function will become highly non-convex with respect to unknown variables, and the optimization process will become very unstable. As a result, the loss function will easily get converged to a local minima with bad solutions.

Instead of minimizing the original loss directly, we try to find a tight upper bound of the loss, which could be minimized more easier. This idea is motivated by re-weighted method and auxiliary function used in non-negative matrix factorization [80], structural sparsity learning [81], graph embedding [82], etc. The theory explains how minimizing the tight upper bound leads to the decrease of the original loss.

Thus, we derive tight upper bounds (i.e. L_1-/L_{21}-norm penalty based robust L_1-PCA model) for the proposed reconstruction loss. Besides, we prove the underlying connection between robustness and regularization. Even though lots of research works show that the regularization helps improving the robustness of machine learning models, our work is the first theoretical proof from a robust point of view in the literature. Towards minimizing the derived tight upper bounds efficiently, we first introduce an augmented Lagrangian multiplier (ALM) based optimization algorithm in matrix-based fashion. Next, we introduce an "exact solver" based optimization algorithm to further improve the robustness of the reconstructed results.

Extensive experimental results on benchmark dataset show that our robust L_1-PCA model obtains better performances in data reconstruction than state-of-the-arts. The proposed robust L_1-PCA model not only learns a low-rank subspace to capture the intrinsic structure of the noisy input data, but also reconstructs the original clean data with a good quality.

5.2 Mathematical Formulation of Robust L_1-PCA Model

In RPCA model, Wright et al of [52] explicitly introduces an additive error term $\mathbf{E}$. Thus, the noisy input data $\mathbf{X}$ is decomposed into the summation of the reconstruction $\mathbf{A}$ and the error $\mathbf{E}$, i.e., $\mathbf{X} = \mathbf{A} + \mathbf{E}$.

Instead of modeling the underlying noise in the original feature space, we propose a novel robust L_1-PCA Model to model additive errors $\delta\mathbf{U}$ and $\delta\mathbf{V}$ on $\mathbf{U}$ (principal directions) and $\mathbf{V}$ (principal components). That is to say, we aims at modeling the noises in lower-dimensional latent subspace, so as to achieve robust solutions of $\mathbf{U}$ and $\mathbf{V}$.

With respect to $\mathbf{U}$, $\delta\mathbf{U}$, $\mathbf{V}$, $\delta\mathbf{V}$, the robust L_1-PCA model is defined as

$$\begin{aligned} \min_{\mathbf{U},\delta\mathbf{U},\mathbf{V},\delta\mathbf{V}} \quad & \left\| \mathbf{X} - (\mathbf{U}+\delta\mathbf{U})(\mathbf{V}+\delta\mathbf{V})^T \right\|_1 \\ \text{s.t.} \quad & \|\delta\mathbf{u}_1\|_1 \le \alpha_1, \ \cdots, \ \|\delta\mathbf{u}_k\|_1 \le \alpha_k \\ & \|\delta\mathbf{v}_1\|_1 \le \beta_1, \ \cdots, \ \|\delta\mathbf{v}_k\|_1 \le \beta_k \end{aligned} \tag{5.8}$$

where $\delta\mathbf{u}_i$ and $\delta\mathbf{v}_i$ are i-th column of corresponding matrices $\delta\mathbf{U}$ and $\delta\mathbf{V}$, α_i and β_i are penalty parameters to control the magnitude of $\|\delta\mathbf{u}_i\|_1$ and $\|\delta\mathbf{v}_i\|_1$. Since the underlying noise is assumed to be sparse and bounded, here ℓ_1-norm is enforced on additive error terms $\delta\mathbf{u}_i$ and $\delta\mathbf{v}_i$.

As discussed before, optimizing four unknown variables $\mathbf{U}$, $\delta\mathbf{U}$, $\mathbf{V}$, $\delta\mathbf{V}$ in problem (5.8) simultaneously is very unstable, especially using the alternative minimization technique. Thus, we derive two tight upper bounds for problem (5.8): (i) L_1-norm penalty based robust L_1-PCA model; (ii) L_{21}-norm penalty based robust L_1-PCA model. Details of the derivation are explained as follows.

5.2.1 L_1-norm Penalty Based Robust L_1-PCA Model

To simplify the optimization, we transform the loss function defined in Eq. (5.8) as the following. Firstly, the triangularity of L_1-norm is applied to obtain a upper bound of the original L_1-norm based loss function

$$\begin{aligned} & \left\| \mathbf{X} - (\mathbf{U}+\delta\mathbf{U})(\mathbf{V}+\delta\mathbf{V})^T \right\|_1 \\ = \quad & \left\| (\mathbf{X}-\mathbf{U}\mathbf{V}^T) - \delta\mathbf{U}\mathbf{V}^T - \mathbf{U}\delta\mathbf{V}^T - \delta\mathbf{U}\delta\mathbf{V}^T \right\|_1 \\ \le \quad & \left\| \mathbf{X}-\mathbf{U}\mathbf{V}^T \right\|_1 + \left\| \delta\mathbf{U}\mathbf{V}^T \right\|_1 + \left\| \mathbf{U}\delta\mathbf{V}^T \right\|_1 + \left\| \delta\mathbf{U}\delta\mathbf{V}^T \right\|_1. \end{aligned} \tag{5.9}$$

Then, we derive a upper bound for the second term $||\delta\mathbf{U}\mathbf{V}^T||_1$ in the right-hand-side of Eq. (5.9)

$$\begin{aligned}
\left\|\delta\mathbf{U}\mathbf{V}^T\right\|_1 &= \sum_{ij}\left|\left(\delta\mathbf{U}\mathbf{V}^T\right)_{ij}\right| \\
&= \sum_{ij}\left|\sum_h \delta\mathbf{U}_{ih}\mathbf{V}_{jh}\right| \\
&\leq \sum_{ij}\sum_h |\delta\mathbf{U}_{ih}|\,|\mathbf{V}_{jh}| \\
&= \sum_h\left(\sum_i |\delta\mathbf{U}_{ih}|\right)\left(\sum_j |\mathbf{V}_{jh}|\right) \qquad (5.10)\\
&= \sum_h \|\delta\mathbf{u}_h\|_1\|\mathbf{v}_h\|_1 \\
&\leq \lambda\left(\sum_h \|\mathbf{v}_h\|_1\right) \\
&= \lambda\|\mathbf{V}\|_1,
\end{aligned}$$

where $\delta\mathbf{u}_h$ and $\mathbf{v}_h$ represent the h-th column of the corresponding matrices $\delta\mathbf{U}$ and $\mathbf{V}$, respectively. The first inequality in Eq. (5.10) is obtained according to the triangularity of L_1-norm, i.e.,

$$\left\|\mathbf{a}^T\mathbf{b}\right\|_1 = \left|\sum_i a_i b_i\right| \leq \sum_i |a_i b_i| = \sum_i |a_i||b_i|. \qquad (5.11)$$

The second inequality is based on the following condition

$$\|\delta\mathbf{u}_h\|_1 \leq \lambda, \qquad (5.12)$$

where we assume the magnitude of additive noise $\delta\mathbf{u}_h$ (which is enforced on principal direction $\mathbf{u}_h$, for $h = 1, 2, \cdots, k$) is explicitly controlled by the same hyperparameter λ, i.e., $\alpha_1 = \cdots = \alpha_k = \lambda$.

For a more complex setting, we can assume the level of each additive noise is controlled by different hyperparameter (i.e. $\lambda_1, \lambda_2, \cdots, \lambda_k$). However, it requires a lot of training time to grid-search optimal hyperparameters in a high-dimensional space.

Besides, data reconstruction is an unsupervised learning task that we don't have any prior knowledge that which principal direction should have a lower level of noise, while which principal direction should have a higher level of noise. For the generality reason, thus we treat the level of additive noise in each principal direction equally by using the same hyperparameter λ.

Via the same above-mentioned procedures, the upper bounds of the third and fourth terms in the right-hand-side of Eq. (5.9) are derived as

$$\left\|\mathbf{U}\delta\mathbf{V}^T\right\|_1 \leq \lambda\|\mathbf{U}\|_1, \tag{5.13}$$

where the inequality is based on the assumption that $\|\delta\mathbf{v}_h\|_1 \leq \lambda$, and

$$\left\|\delta\mathbf{U}\delta\mathbf{V}^T\right\|_1 \leq k\lambda^2, \tag{5.14}$$

where the inequality is based on the assumptions that $\|\delta\mathbf{u}_h\|_1 \leq \lambda$ and $\|\delta\mathbf{v}_h\|_1 \leq \lambda$, i.e., $\alpha_1 = \cdots = \alpha_k = \lambda$, $\beta_1 = \cdots = \beta_k = \lambda$. Without loss of the generality, here we also treat the level of additive nose in each principal component equally.

Combining the results in Eq. (5.10), Eq. (5.13), Eq. (5.14) together, finally the tight upper bound of problem (5.8) is obtained as

$$\begin{aligned} & \left\|\mathbf{X} - (\mathbf{U} + \delta\mathbf{U})(\mathbf{V} + \delta\mathbf{V})^T\right\|_1 \\ \leq \quad & \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 + \lambda\|\mathbf{U}\|_1 + \lambda\|\mathbf{V}\|_1 + k\lambda^2. \end{aligned} \tag{5.15}$$

Since the last term of Eq. (5.15) is irrelevant to the optimization of $\mathbf{U}$ and $\mathbf{V}$, after ignoring this constant, the robust L_1-PCA model defined in Eq. (5.8) is simplified to minimize the following upper bound

$$\min_{\mathbf{U},\mathbf{V}} \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 + \lambda\|\mathbf{U}\|_1 + \lambda\|\mathbf{V}\|_1, \tag{5.16}$$

where λ is the hyperparameter to control the level of sparsity in principal directions $\mathbf{U}$ and principal components $\mathbf{V}$.

Thus it can be seen that adding noises $\delta\mathbf{U}$ and $\delta\mathbf{V}$ into L_1-PCA model are equivalent to enforcing sparsity-induced L_1-norm on corresponding variables $\mathbf{U}$ and $\mathbf{V}$. That is to say, the underlying connection between robustness and regularization is built through our derivation. From a robust point of view, we theoretically prove the well-known fact that the regularization helps improving the robustness of machine learning models.

5.2.2 L_{21}-norm Penalty Based Robust L_1-PCA Model

With different assumption on additive noises, we can derive another tight upper bound of the second term in the right-hand-side of Eq. (5.9) as follow

$$\begin{aligned}
\left\|\delta\mathbf{U}\mathbf{V}^T\right\|_1 &= \left\|\sum_h \delta\mathbf{u}_h\mathbf{v}_h^T\right\|_1 \\
&\leq \sum_h \left\|\delta\mathbf{u}_h\mathbf{v}_h^T\right\|_1 \\
&= \sum_h \sum_{j=1}^{d} \left\|\left\langle \delta u_h^j \cdot \vec{\mathbf{1}}, \mathbf{v}_h\right\rangle\right\|_1 \\
&\leq \sum_h \sum_{j=1}^{d} \left(\sqrt{n}\left|\delta u_h^j\right|\right)\left(\|\mathbf{v}_h\|_2\right) \qquad (5.17)\\
&= \sum_h \sqrt{n}\ \|\delta\mathbf{u}_h\|_1\|\mathbf{v}_h\|_2 \\
&\leq \left(\sqrt{n}\frac{\lambda}{\sqrt{n}}\right)\left(\sum_h \|\mathbf{v}_h\|_2\right) \\
&= \lambda\|\mathbf{V}\|_{21},
\end{aligned}$$

where $\delta\mathbf{u}_h$ and $\mathbf{v}_h$ are the h-th column of corresponding matrices $\delta\mathbf{U}$ and $\mathbf{V}$ respectively, δu_h^j is the j-th element in column vector $\delta\mathbf{u}_h$, and $\vec{\mathbf{1}}$ is the vector with n elements and all ones. The first inequality in Eq. (5.17) is obtained according to the triangularity of L_1-norm, i.e.,

$$\left\|\mathbf{a}_1^T\mathbf{b}_1 + \cdots + \mathbf{a}_n^T\mathbf{b}_n\right\|_1 \leq \left\|\mathbf{a}_1^T\mathbf{b}_1\right\| + \cdots + \left\|\mathbf{a}_n^T\mathbf{b}_n\right\|_1. \qquad (5.18)$$

The second inequality in Eq. (5.17) is obtained using Cauchy Schwarz inequality, i.e., given $\mathbf{x}$ and $\mathbf{y}$, we have the following inequality in dot-product space

$$|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\|, \tag{5.19}$$

where $\mathbf{x} = \delta u_h^j \cdot \vec{\mathbf{1}}$ and $\mathbf{y} = \mathbf{v}_h$. The third inequality is based on the following condition

$$\|\delta \mathbf{u}_h\|_1 \leq \frac{\lambda}{\sqrt{n}}, \tag{5.20}$$

where we assume the magnitude of additive noise $\delta \mathbf{u}_h$ (which is enforced on principal direction $\mathbf{u}_h$, for $h = 1, 2, \cdots, k$) is explicitly controlled by the same hyperparameter $\lambda/\sqrt{n}$, i.e., $\alpha_1 = \cdots = \alpha_k = \lambda/\sqrt{n}$.

By using the same above-mentioned procedures, the upper bounds of the third and fourth terms in the right-hand-side of Eq. (5.9) are derived as

$$\left\|\mathbf{U} \delta \mathbf{V}^T\right\|_1 \leq \lambda \|\mathbf{U}\|_{21}, \tag{5.21}$$

where the inequality is based on the assumption that $\|\delta \mathbf{v}_h\|_1 \leq \frac{\lambda}{\sqrt{d}}$, and

$$\left\|\delta \mathbf{U} \delta \mathbf{V}^T\right\|_1 \leq \frac{k\lambda^2}{\sqrt{nd}}, \tag{5.22}$$

where the inequality is based on the assumptions that $\|\delta \mathbf{u}_h\|_1 \leq \frac{\lambda}{\sqrt{n}}$ and $\|\delta \mathbf{v}_h\|_1 \leq \frac{\lambda}{\sqrt{d}}$, i.e., $\alpha_1 = \cdots = \alpha_k = \lambda/\sqrt{n}$, $\beta_1 = \cdots = \beta_k = \lambda/\sqrt{d}$. Without loss of the generality, here we also treat the level of additive nose in each principal component equally.

Combining the results in Eq. (5.17), Eq. (5.21), Eq. (5.22) together, another tight upper bound of problem (5.8) is obtained as

$$\begin{aligned} & \left\|\mathbf{X} - (\mathbf{U} + \delta \mathbf{U}) (\mathbf{V} + \delta \mathbf{V})^T\right\|_1 \\ \leq \quad & \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 + \lambda \|\mathbf{U}\|_{21} + \lambda \|\mathbf{V}\|_{21} + \tfrac{k\lambda^2}{\sqrt{nd}}. \end{aligned} \tag{5.23}$$

Ignoring the last constant term, the robust L_1-PCA model defined in Eq. (5.8) is simplified to minimize the following upper bound

$$\min_{\mathbf{U}, \mathbf{V}} \ \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 + \lambda \|\mathbf{U}\|_{21} + \lambda \|\mathbf{V}\|_{21}, \tag{5.24}$$

where λ is the hyperparameter to control the level of sparsity in principal directions $\mathbf{U}$ and principal components $\mathbf{V}$.

Thus, adding noises $\delta\mathbf{U}$ and $\delta\mathbf{V}$ into L_1-PCA model are equivalent to enforcing sparsity-induced L_{21}-norm on corresponding variables $\mathbf{U}$ and $\mathbf{V}$.

With different assumptions on additive noises of principal directions and principal components, we derived two tight upper bounds for robust L_1-PCA model. For L_1-norm penalty based robust L_1-PCA model, the magnitude of additive noise is controlled by λ. For L_{21}-norm penalty based robust L_1-PCA model, the magnitude of additive noise is controlled by λ and its length.

5.3 Augmented Lagrangian Multiplier Based Optimization Algorithm

In the previous section, we derive two tight upper bounds for robust L_1-PCA models in the form of L_1-norm penalty and L_{21}-norm penalty, see Eq. (5.16) and Eq. (5.24). As it can be seen in the derived tight upper bounds, we only have two unknown variables and one hyperparameter, which greatly simplifies the optimization of the original problem (5.8).

Towards minimizing the L_1-/L_{21}-norm penalty based robust L_1-PCA models, in the following, we propose an augmented Lagrangian multiplier based optimization algorithm, which can work efficiently with proximal operators such as L_1-norm and L_{21}-norm regularizations. All the computations can be implemented in matrix-based fashion.

5.3.1 Efficient Optimization Algorithm for Solving L_1-norm Penalty Based Robust L_1-PCA Model

As it can be seen in Eq. (5.16), we have a multivariate optimization problem, where two unknown variables $\mathbf{U}$, $\mathbf{V}$ need to be minimized. The first idea that comes into the mind is using the alternative optimization method. However, computing the derivative with respect to one variable while fixing another variable becomes impractical, since those three L_1-norm based terms in the objective function are non-smooth and non-differentiable.

To resolve this difficulty, we apply augmented Lagrangian multiplier [43] to solve the optimization problem (5.16). First, we introduce three auxiliary variables **E**, **F**, and **G** to make the optimization separable between loss and penalty. Thus, the original optimization problem (5.16) becomes to

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{F},\mathbf{G}} \quad & \|\mathbf{E}\|_1 + \lambda\|\mathbf{F}\|_1 + \lambda\|\mathbf{G}\|_1 \\ \text{s.t.} \quad & \mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} = 0, \\ & \mathbf{U} - \mathbf{F} = 0, \\ & \mathbf{V} - \mathbf{G} = 0. \end{aligned} \tag{5.25}$$

Even though the original unconstrained problem is transformed into an constrained problem, the optimization becomes much more easier. Before, the optimization with respect to **U** or **V** are in both L_1 loss and L_1 regularization, which is hard to solve. After adding auxiliary variables, the optimization with respect to **U** or **V** are decomposed into independent subproblems.

Then, we introduce three Lagrangian multipliers $\mathbf{\Omega}$, $\mathbf{\Sigma}$, and $\mathbf{\Lambda}$ to enforce the equality constraints in problem (5.25) explicitly. Thus, the Lagrangian function is defined as

$$\begin{aligned} \mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{F},\mathbf{G},\mathbf{\Omega},\mathbf{\Sigma},\mathbf{\Lambda},\mu) \;=\; & \|\mathbf{E}\|_1 + \lambda\|\mathbf{F}\|_1 + \lambda\|\mathbf{G}\|_1 \\ & + \left\langle \mathbf{\Omega}, \mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} \right\rangle + \tfrac{\mu}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E}\right\|_F^2 \\ & + \langle \mathbf{\Sigma}, \mathbf{U} - \mathbf{F} \rangle + \tfrac{\mu}{2}\|\mathbf{U} - \mathbf{F}\|_F^2 \\ & + \langle \mathbf{\Lambda}, \mathbf{V} - \mathbf{G} \rangle + \tfrac{\mu}{2}\|\mathbf{V} - \mathbf{G}\|_F^2 \end{aligned} \tag{5.26}$$

where $\langle\cdot,\cdot\rangle$ is inner product, i.e., $\langle\mathbf{A},\mathbf{B}\rangle = \mathrm{tr}(\mathbf{A}^T\mathbf{B})$, μ is penalty parameter, and $\mathbf{\Omega}$, $\mathbf{\Sigma}$, $\mathbf{\Lambda}$ are Lagrangian multipliers.

Eq. (5.26) can be further rewritten to the following equivalent formulation

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{F},\mathbf{G}} \quad & \|\mathbf{E}\|_1 + \lambda\|\mathbf{F}\|_1 + \lambda\|\mathbf{G}\|_1 \\ & + \tfrac{\mu}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} + \tfrac{\mathbf{\Omega}}{\mu}\right\|_F^2 \\ & + \tfrac{\mu}{2}\left\|\mathbf{U} - \mathbf{F} + \tfrac{\mathbf{\Sigma}}{\mu}\right\|_F^2 \\ & + \tfrac{\mu}{2}\left\|\mathbf{V} - \mathbf{G} + \tfrac{\mathbf{\Lambda}}{\mu}\right\|_F^2 .\end{aligned} \tag{5.27}$$

As compared to the original optimization problem (5.16), the optimization with respect to three L_1-norm based terms are decoupled in problem (5.27). Thus, we can apply the alternative minimization method to solve problem (5.27), which can be decomposed to several subproblems with respect to one unknown variable while fixing the rest of unknown variables.

Details of the optimization with respect to each subproblem are explained in the following.

5.3.1.1 Solving for E, F, G

The optimization with respect to **E**, **F**, **G** is decomposed as follows:

(i) Fixing **F**, **G**, **U**, **V**, the optimization with respect to **E** becomes to

$$\underset{\mathbf{E}}{min}\ \|\mathbf{E}\|_1 + \frac{\mu}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} + \frac{\mathbf{\Omega}}{\mu}\right\|. \tag{5.28}$$

(ii) Fixing **E**, **G**, **U**, **V**, the optimization with respect to **F** becomes to

$$\underset{\mathbf{F}}{min}\ \lambda\|\mathbf{F}\|_1 + \frac{\mu}{2}\left\|\mathbf{U} - \mathbf{F} + \frac{\mathbf{\Sigma}}{\mu}\right\|_F^2. \tag{5.29}$$

(iii) Fixing **E**, **F**, **U**, **V**, the optimization with respect to **G** becomes to

$$\underset{\mathbf{G}}{min}\ \lambda\|\mathbf{G}\|_1 + \frac{\mu}{2}\left\|\mathbf{V} - \mathbf{G} + \frac{\mathbf{\Lambda}}{\mu}\right\|_F^2. \tag{5.30}$$

All the three subproblems are L_1-norm based proximal operator-type problem. Thus, we first introduce a general optimization problem with respect to L_1-norm, which then can be applied to solve each subproblem in a similar way.

Theorem 5.3.1. *Given the L_1-norm based proximal operator-type problem*

$$\min_{\mathbf{B}} \ \alpha\|\mathbf{B}\|_1 + \frac{1}{2}\|\mathbf{B} - \mathbf{A}\|_F^2, \tag{5.31}$$

the optimal solution $\mathbf{B}^*$ of Eq. (5.31) is given by

$$\mathbf{B}_{ij}^* = \text{sign}(\mathbf{A}_{ij}) \cdot \max\left(|\mathbf{A}_{ij}| - \alpha, 0\right). \tag{5.32}$$

Proof of Theorem 5.3.1. If the entries in matrix $\mathbf{B}$ are not correlated, Eq. (5.31) can be simplified to the optimization problem with respect to a single entry

$$\min_{b} \ \alpha|b| + \frac{1}{2}(b - a)^2.$$

Since $\text{sign}(b) = \text{sign}(a)$, the objective function becomes to

$$\min_{b} \ \alpha|b| + \frac{1}{2}(|b| - |a|)^2.$$

By setting derivative with respect to $|b|$ to zero, we have $|b^*| - |a| + \alpha = 0$. Thus, the optimal solution is obtained as $b^* = \text{sign}(a) \cdot \max(|a| - \alpha, 0)$. By setting $a = A_{ij}$ and $b^* = B_{ij}^*$, we have the same optimal solution given in Eq. (5.32). □

Reorganizing the terms in Eqs. (5.28-5.30), the subproblem with respect to $\mathbf{E}$, $\mathbf{F}$, $\mathbf{G}$ can be reduced to the same mathematical formulation defined in Eq. (5.31). Thus, by using Theorem 5.3.1, the optimal solutions of $\mathbf{E}^*$, $\mathbf{F}^*$, $\mathbf{G}^*$ are obtained as follows

$$\mathbf{E}_{ij}^* = \text{sign}\left(\widetilde{\mathbf{X}}_{ij}\right) \cdot \max\left(\left|\widetilde{\mathbf{X}}_{ij}\right| - \frac{1}{\mu}, 0\right), \tag{5.33}$$

$$\mathbf{F}_{ij}^* = \text{sign}\left(\widetilde{\mathbf{U}}_{ij}\right) \cdot \max\left(\left|\widetilde{\mathbf{U}}_{ij}\right| - \frac{\lambda}{\mu}, 0\right), \tag{5.34}$$

$$\mathbf{G}_{ij}^* = \text{sign}\left(\widetilde{\mathbf{V}}_{ij}\right) \cdot \max\left(\left|\widetilde{\mathbf{V}}_{ij}\right| - \frac{\lambda}{\mu}, 0\right), \tag{5.35}$$

where $\widetilde{\mathbf{X}} = \mathbf{X} - \mathbf{U}\mathbf{V}^T + \mathbf{\Omega}/\mu$, $\widetilde{\mathbf{U}} = \mathbf{U} + \mathbf{\Sigma}/\mu$, $\widetilde{\mathbf{V}} = \mathbf{V} + \mathbf{\Lambda}/\mu$.

5.3.1.2 Solving for $\mathbf{U}$, $\mathbf{V}$

The optimization with respect to $\mathbf{U}$, $\mathbf{V}$ is decomposed as follows:

(i) Fixing $\mathbf{E}$, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{V}$, the optimization with respect to $\mathbf{U}$ becomes to

$$\min_{\mathbf{U}} \frac{\mu}{2}\left\|\widetilde{\mathbf{X}} - \mathbf{U}\mathbf{V}^T\right\|_F^2 + \frac{\mu}{2}\|\mathbf{U} - \mathbf{C}\|_F^2, \tag{5.36}$$

where $\widetilde{\mathbf{X}} = \mathbf{X} - \mathbf{E} + \mathbf{\Omega}/\mu$, $\mathbf{C} = \mathbf{F} - \mathbf{\Sigma}/\mu$.

(ii) Fixing $\mathbf{E}$, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{U}$, the optimization with respect to $\mathbf{V}$ becomes to

$$\min_{\mathbf{V}} \frac{\mu}{2}\left\|\widetilde{\mathbf{X}} - \mathbf{U}\mathbf{V}^T\right\|_F^2 + \frac{\mu}{2}\|\mathbf{V} - \mathbf{D}\|_F^2, \tag{5.37}$$

where $\widetilde{\mathbf{X}} = \mathbf{X} - \mathbf{E} + \mathbf{\Omega}/\mu$, $\mathbf{D} = \mathbf{G} - \mathbf{\Lambda}/\mu$.

As it can be seen that all the terms in Eq. (5.36) and Eq. (5.37) are smooth and differentiable, the derivative can be computed directly. By setting derivatives of Eq. (5.36) and Eq. (5.37) with respect to $\mathbf{U}$ and $\mathbf{V}$ to zero respectively, we obtain the optimal solutions of $\mathbf{U}^*$ and $\mathbf{V}^*$

$$\mathbf{U}^* = \left(\mathbf{C} + \widetilde{\mathbf{X}}\mathbf{V}\right)\left(\mathbf{V}^T\mathbf{V} + \mathbf{I}\right)^{-1}, \tag{5.38}$$

$$\mathbf{V}^* = \left(\mathbf{D} + \widetilde{\mathbf{X}}^T\mathbf{U}\right)\left(\mathbf{U}^T\mathbf{U} + \mathbf{I}\right)^{-1}, \tag{5.39}$$

where $\mathbf{I}$ is k-by-k identity matrix.

5.3.1.3 Updating Parameters: $\mathbf{\Omega}$, $\mathbf{\Sigma}$, $\mathbf{\Lambda}$, μ

Finally, we update parameters $\mathbf{\Omega}$, $\mathbf{\Sigma}$, $\mathbf{\Lambda}$, ν at the end of each iteration as the following

$$\mathbf{\Omega} \Leftarrow \mathbf{\Omega} + \mu \cdot \left(\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E}\right), \tag{5.40}$$

$$\mathbf{\Sigma} \Leftarrow \mathbf{\Sigma} + \mu \cdot (\mathbf{U} - \mathbf{F}), \tag{5.41}$$

$$\mathbf{\Lambda} \Leftarrow \mathbf{\Lambda} + \mu \cdot (\mathbf{V} - \mathbf{G}), \tag{5.42}$$

$$\mu \Leftarrow \rho \cdot \mu. \tag{5.43}$$

5.3.2 Efficient Optimization Algorithm for Solving L_{21}-norm Penalty Based Robust L_1-PCA Model

Since L_{21}-norm shares the same non-smooth and non-differentiable property as L_1-norm, we also apply aforementioned ALM method to solve the L_{21}-norm penalty based robust L_1-PCA model.

First, we introduce three auxiliary variables $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{G}$ to make the optimization separable between L_1-norm based loss and L_{21}-norm based regularization. Thus, the original optimization problem (5.24) becomes to

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{F},\mathbf{G}} \quad & \|\mathbf{E}\|_1 + \lambda\|\mathbf{F}\|_{21} + \lambda\|\mathbf{G}\|_{21} \\ \text{s.t.} \quad & \mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} = 0, \\ & \mathbf{U} - \mathbf{F} = 0, \\ & \mathbf{V} - \mathbf{G} = 0. \end{aligned} \tag{5.44}$$

By using ALM method, Eq. (5.44) can be further rewritten as

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{E},\mathbf{F},\mathbf{G}} \quad & \|\mathbf{E}\|_1 + \lambda\|\mathbf{F}\|_{21} + \lambda\|\mathbf{G}\|_{21} \\ & + \tfrac{\mu}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T - \mathbf{E} + \tfrac{\boldsymbol{\Omega}}{\mu}\right\|_F^2 \\ & + \tfrac{\mu}{2}\left\|\mathbf{U} - \mathbf{F} + \tfrac{\boldsymbol{\Sigma}}{\mu}\right\|_F^2 \\ & + \tfrac{\mu}{2}\left\|\mathbf{V} - \mathbf{G} + \tfrac{\boldsymbol{\Lambda}}{\mu}\right\|_F^2, \end{aligned} \tag{5.45}$$

where $\boldsymbol{\Omega}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Lambda}$ are Lagrangian multipliers, and μ is the penalty parameter.

Thus it can be seen that the difference between problem (5.27) and problem (5.45) is only the optimization with respect to $\mathbf{F}$ and $\mathbf{G}$. The optimal solution of variables $\mathbf{E}$, $\mathbf{U}$, $\mathbf{V}$ and the update of parameters $\boldsymbol{\Omega}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Lambda}$, μ given in previous section can be reused here to solve the L_{21}-norm penalty based Robust L_1-PCA model.

5.3.2.1 Solving for $\mathbf{E}$

Same as Eq. (5.33).

5.3.2.2 Solving for $\mathbf{F}$, $\mathbf{G}$

The optimization with respect to $\mathbf{F}$, $\mathbf{G}$ is decomposed as follows:

(i) Fixing $\mathbf{E}$, $\mathbf{G}$, $\mathbf{U}$, $\mathbf{V}$, the optimization with respect to $\mathbf{F}$ becomes to

$$\min_{\mathbf{F}} \ \lambda\|\mathbf{F}\|_{21} + \frac{\mu}{2}\left\|\mathbf{U} - \mathbf{F} + \frac{\mathbf{\Sigma}}{\mu}\right\|_F^2. \tag{5.46}$$

(ii) Fixing $\mathbf{E}$, $\mathbf{F}$, $\mathbf{U}$, $\mathbf{V}$, the optimization with respect to $\mathbf{G}$ becomes to

$$\min_{\mathbf{G}} \ \lambda\|\mathbf{G}\|_{21} + \frac{\mu}{2}\left\|\mathbf{V} - \mathbf{G} + \frac{\mathbf{\Lambda}}{\mu}\right\|_F^2. \tag{5.47}$$

As compared to previous subsection, the regularization enforced on $\mathbf{F}$ and $\mathbf{G}$ has been changed from L_1-norm to L_{21}-norm. Now, these two subproblems become to the L_{21}-norm based proximal operator-type problem. Thus, we first introduce a general optimization problem with respect to L_{21}-norm, which then can be applied to solve each subproblem in a similar way.

Theorem 5.3.2. *Given the L_{21}-norm based proximal operator-type problem*

$$\min_{\mathbf{B}} \ \alpha\|\mathbf{B}\|_{21} + \frac{1}{2}\|\mathbf{B} - \mathbf{A}\|_F^2, \tag{5.48}$$

the optimal solution $\mathbf{B}^*$ of Eq. (5.48) is given by

$$\mathbf{b}_i = \mathbf{a}_i \cdot \max\left(1 - \frac{\alpha}{\|\mathbf{a}_i\|_2}, 0\right), \tag{5.49}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are i-th column of the corresponding matrices $\mathbf{A}$ and $\mathbf{B}$ respectively.

Proof of Theorem 5.3.2. If the columns in matrix $\mathbf{B}$ are not correlated, Eq. (5.48) can be simplified to the optimization problem with respect to a single column

$$\min_{\mathbf{x}} \ \alpha\|\mathbf{x}\|_2 + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2.$$

Since $\mathbf{x} = \rho\mathbf{y}$ ($\rho \in \mathbb{R}^+$), the objective function becomes to

$$\min_{\rho} \ \alpha\rho\|\mathbf{y}\|_2 + \frac{(\rho-1)^2}{2}\|\mathbf{y}\|_2^2.$$

By setting derivative with respect to ρ to zero, we have $\alpha\|\mathbf{y}\|_2 + (\rho - 1)\|\mathbf{y}\|_2^2 = 0$. Thus, the optimal solution of ρ is $\rho^* = \max(1-\alpha/\|\mathbf{y}\|_2, 0)$. According to the relation $\mathbf{x}^* = \rho^*\mathbf{y}$, the optimal solution of $\mathbf{x}$ is obtained as $\mathbf{x}^* = \mathbf{y} \cdot \max(1-\alpha/\|\mathbf{y}\|_2, 0)$. Thus, the optimal solution defined in Eq. (5.49) is obtained by setting $\mathbf{x}^* = \mathbf{a}_i$ and $\mathbf{y} = \mathbf{b}_i$, where $\mathbf{a}_i$ and $\mathbf{b}_i$ are i-th column of matrices $\mathbf{A}$ and $\mathbf{B}$ respectively. □

Reorganizing the terms in Eqs. (5.46-5.47), the subproblem with respect to $\mathbf{F}$ and $\mathbf{G}$ can be reduced to the same mathematical formulation defined in Eq. (5.48). Thus, by using Theorem 5.3.2, we obtain the optimal solutions of $\mathbf{F}^*$ and $\mathbf{G}^*$ as the following

$$\mathbf{f}_i^* = \widetilde{\mathbf{u}}_i \cdot \max\left(1 - \frac{\lambda/\mu}{\|\widetilde{\mathbf{u}}_i\|_2}, 0\right), \tag{5.50}$$

$$\mathbf{g}_i^* = \widetilde{\mathbf{v}}_i \cdot \max\left(1 - \frac{\lambda/\mu}{\|\widetilde{\mathbf{v}}_i\|_2}, 0\right), \tag{5.51}$$

where $\mathbf{f}_i^*$, $\mathbf{g}_i^*$, $\widetilde{\mathbf{u}}_i$, $\widetilde{\mathbf{v}}_i$ are i-th column of the corresponding matrices $\mathbf{F}^*$, $\mathbf{G}^*$, $\widetilde{\mathbf{U}}$, $\widetilde{\mathbf{V}}$ respectively, $\widetilde{\mathbf{U}} = \mathbf{U} + \mathbf{\Sigma}/\mu$, and $\widetilde{\mathbf{V}} = \mathbf{V} + \mathbf{\Lambda}/\mu$.

5.3.2.3 Solving for $\mathbf{U}$, $\mathbf{V}$

Same as Eqs. (5.38-5.39).

5.3.2.4 Updating Parameters: $\mathbf{\Omega}$, $\mathbf{\Sigma}$, $\mathbf{\Lambda}$, μ

Same as Eqs. (5.40-5.43).

5.3.3 Implementation Details of ALM Based Optimization Algorithm

The complete algorithm for solving L_1-/L_{21}-norm penalty based robust L_1-PCA models is summarized in Algorithm 4. Since the proposed ALM based algorithm uses alternative optimization technique, the optimal solution of $\mathbf{U}$ and $\mathbf{V}$ is obtained iteratively until the objective function J defined in Eq. (5.16) and Eq. (5.24) converges.

Algorithm 4 ALM Based Optimization Algorithm for Solving L_1-/L_{21}-norm Penalty Based Robust L_1-PCA Models.

```
Input: data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$, rank $k$, hyperparameter $\lambda$.
Output: principal directions $\mathbf{U} \in \mathbb{R}^{d\times k}$, principal components $\mathbf{V} \in \mathbb{R}^{n\times k}$.
Parameter settings:
    $\mathbf{E} = 0$, $\mathbf{F} = 0$, $\mathbf{G} = 0$, $\mathbf{\Omega} = 0$, $\mathbf{\Sigma} = 0$, $\mathbf{\Lambda} = 0$.
    $t = 0$, $\rho = 1.1$, $\mu = 1/\sum_{ij}\mathbf{X}_{ij}^2$.
Initialization:
    (i) SVD initialization: $\mathbf{U}^{(t)}, \mathbf{V}^{(t)} = \mathrm{svd}(\mathbf{X}, k)$;
    (ii) Random initialization: $\mathbf{U}^{(t)} = \mathrm{rand}(p, k)$, $\mathbf{V}^{(t)} = \mathrm{rand}(n, k)$.
repeat
    Computing $\mathbf{E}^{(t+1)}$ using Eq. (5.33)
    (i) For $L_1$-norm penalty, computing $\mathbf{F}^{(t+1)}$ using Eq. (5.34);
    (ii) For $L_{21}$-norm penalty, computing $\mathbf{F}^{(t+1)}$ using Eq. (5.50).
    (i) For $L_1$-norm penalty, computing $\mathbf{G}^{(t+1)}$ using Eq. (5.35);
    (ii) For $L_{21}$-norm penalty, computing $\mathbf{G}^{(t+1)}$ using Eq. (5.51).
    Computing $\mathbf{U}^{(t+1)}$ using Eq. (5.38).
    Computing $\mathbf{V}^{(t+1)}$ using Eq. (5.39).
    Updating $\mathbf{\Omega}^{(t+1)}$ using Eq. (5.40).
    Updating $\mathbf{\Sigma}^{(t+1)}$ using Eq. (5.41).
    Updating $\mathbf{\Lambda}^{(t+1)}$ using Eq. (5.42).
    Updating $\mu^{(t+1)}$ using Eq. (5.43).
    $t = t + 1$.
until Objective function $J$ converges, i.e., $|J_t - J_{t-1}|/J_t < 1e-6$.
return Optimal solutions: $\mathbf{U}^*$, $\mathbf{V}^*$.
```

Regarding the starting point of the optimization, we provide two ways to initialize $\mathbf{U}^{(0)}, \mathbf{V}^{(0)}$: (i) random initialization; (ii) svd initialization. According to our preliminary experiments, optimization using svd initialization usually obtains better result than using random initialization. Besides, optimization using random initialization needs more iterations to converge.

After computing the optimal solution of variables $\mathbf{E}$, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{U}$, $\mathbf{V}$, parameters $\mathbf{\Omega}$, $\mathbf{\Sigma}$, $\mathbf{\Lambda}$, μ are updated at the end of each iteration.

Finally, the proposed optimization algorithm converges when $|J_t - J_{t-1}|/J_t < 1e-6$. That is to say, the reduction in the objective function value of Eq. (5.16) and Eq. (5.24) is sufficiently small.

5.4 Exact Solver Based Optimization Algorithm

Even though it is very efficient for minimizing the derived tight upper bounds, the proposed ALM based optimization algorithm usually has an early stopping problem, i.e., the objective function converges to a bad local minima that the reconstructed results are still noisy. Thus, it requires more training time to finetune the hyperparameters in robust L_1-PCA model to obtain a good solution.

To resolve the above-mentioned problems, we introduce a "exact solver" based optimization algorithm to further improve the robustness of the reconstructed results. In the "exact solver" method, one element of principal directions and principal components is optimized at each time. The global minimum of objective function with respect to a single element can be obtained in linear time.

5.4.1 Expansion of Objective Function

Before introducing the exact solver based algorithm, first we expand both loss and penalty terms in Eq. (5.16) and Eq. (5.24) as the following. Here, we simplify the robust L_1-PCA model with respect to a single entry (i_0, h_0) of matrix $\mathbf{U}$ ($i_0 = 1, \cdots, d$, $h_0 = 1, \cdots, k$), while assuming the rest of the entries ($i \neq i_0$, $h \neq h_0$) are fixed.

The L_1-norm based loss term in Eq. (5.16) and Eq. (5.24) is expanded as

$$\begin{aligned} & J_{loss}(\mathbf{U}, \mathbf{V}) \\ = & \left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 \\ = & \sum_{i \neq i_0} \sum_j \left|\mathbf{X}_{ij} - \sum_h \mathbf{U}_{ih}\mathbf{V}_{jh}\right| + \sum_j \left|\mathbf{X}_{i_0 j} - \sum_{h \neq h_0} \mathbf{U}_{i_0 h}\mathbf{V}_{jh} - \mathbf{U}_{i_0 h_0}\mathbf{V}_{jh_0}\right| \\ = & \sum_j \left|\widetilde{\mathbf{X}}_{i_0 j} - \mathbf{U}_{i_0 h_0}\mathbf{V}_{jh_0}\right| + \text{const}(i \neq i_0), \end{aligned} \tag{5.52}$$

where $\widetilde{\mathbf{X}}_{i_0 j} = \mathbf{X}_{i_0 j} - \sum_{h \neq h_0} \mathbf{U}_{i_0 h}\mathbf{V}_{jh}$.

The L_1-norm based penalty term in Eq. (5.16) is expanded as

$$\begin{aligned} J_{penalty-L_1}(\mathbf{U}) &= \|\mathbf{U}\|_1 \\ &= \left|\mathbf{U}_{i_0 h_0}\right| + \sum_{i \neq i_0} \sum_{h \neq h_0} \left|\mathbf{U}_{ih}\right| \\ &= \left|\mathbf{U}_{i_0 h_0}\right| + \text{const}(i \neq i_0, h \neq h_0). \end{aligned} \tag{5.53}$$

The L_{21}-norm based penalty term in Eq. (5.24) is expanded as

$$\begin{aligned} J_{penalty-L_{21}}(\mathbf{U}) &= \|\mathbf{U}\|_{21} \\ &= \left(\mathbf{U}_{i_0 h_0}^2 + \sum_{i \neq i_0} \mathbf{U}_{ih_0}^2\right)^{0.5} + \sum_{h \neq h_0} \left(\sum_i \mathbf{U}_{ih}^2\right)^{0.5} \\ &= \left(\mathbf{U}_{i_0 h_0}^2 + c\right)^{0.5} + \text{const}(h \neq h_0), \end{aligned} \tag{5.54}$$

where $c = \sum_{i \neq i_0} \mathbf{U}_{ih_0}^2$.

Via transposing L_1-norm based loss term as $\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\right\|_1 = \left\|\mathbf{X}^T - \mathbf{V}\mathbf{U}^T\right\|_1$, then we can apply above procedure to decompose the objective function with respect to a single entry (j_0, h_0) of matrix $\mathbf{V}$ ($j_0 = 1, \cdots, n$, $h_0 = 1, \cdots, k$), while assuming the rest of the entries ($j \neq j_0$, $h \neq h_0$) are fixed.

5.4.2 Minimization of Objective Function

After the expansion, we can minimize the simplified objective function with respect to a single element at each time.

Combining the results in Eqs. (5.52-5.54) together, the L_1-norm penalty based robust L_1-PCA model defined in Eq. (5.16) becomes to the following optimization problem with respect to u,

$$\min_{u} \; J_1(u) = \left\|\widetilde{\mathbf{x}} - u \cdot \mathbf{v}\right\|_1 + \lambda \left|u\right|, \tag{5.55}$$

where u is the (i_0, h_0)-th entry of the corresponding matrix $\mathbf{U}$, $\widetilde{\mathbf{x}}$ is the transpose of i_0-th row of the corresponding matrix $\widetilde{\mathbf{X}}$, $\mathbf{v}$ is the h_0-th column of corresponding matrix $\mathbf{V}$.

On the other hand, the L_{21}-norm penalty based robust L_1-PCA model defined in Eq. (5.24) becomes to the following optimization problem with respect to u,

$$\min_{u} \; J_{21}(u) = \left\|\widetilde{\mathbf{x}} - u \cdot \mathbf{v}\right\|_1 + \lambda\sqrt{u^2 + c}, \tag{5.56}$$

where u, $\widetilde{\mathbf{x}}$, $\mathbf{v}$ have same definitions as Eq. (5.55), and $c = \sum_{i \neq i_0} \mathbf{U}_{ih_0}^2$.

Next, we will explain the details of the "exact solver" method, which can obtain the global optimal solution u^* of $J_1(u)$ and $J_{21}(u)$, i.e., the optimal solution $\mathbf{U}_{i_0h_0}^*$ of problem (5.16) and problem (5.24), for $i_0 = 1, \cdots, d$, $h_0 = 1, \cdots, k$.

5.4.2.1 Solving for $J_1(u)$

First, we rewrite $J_1(u)$ defined in Eq. (5.55) equivalently as

$$\begin{aligned} J_1(u) &= \left\|\widetilde{\mathbf{x}} - u \cdot \mathbf{v}\right\|_1 + \lambda \left|u\right| \\ &= \sum_{i=1}^{n} \left|v_i\right| \left|\tfrac{\widetilde{x}_i}{v_i} - u\right| + \lambda \left|0 - u\right| \\ &= \sum_{i=0}^{n} \left|b_i\right| \left|a_i - u\right| \end{aligned} \tag{5.57}$$

where $\mathbf{a} = (0, \widetilde{x}_1/v_1, \cdots, \widetilde{x}_n/v_n)^T$, $\mathbf{b} = (\lambda, v_1, \cdots, v_n)^T$.

Next, we compute the gradient of $J_1(u)$ with respect to u as

$$\nabla J_1(u) = \sum_{i=0}^{n} |b_i| \cdot \operatorname{sign}(u - a_i). \tag{5.58}$$

It is obvious that the gradient changes when u is crossing any a_i for $i = 0, \cdots, n$, since $\operatorname{sign}(u - a_i)$ turns from -1 to $+1$ at the endpoint a_i.

For simplicity of the gradient analysis of $J_1(u)$, we sort $\mathbf{a} = (a_0, a_1, \cdots, a_n)^T$ in an ascending order $\mathcal{I}$, where $a_{\mathcal{I}_0} \leq a_{\mathcal{I}_1} \leq a_{\mathcal{I}_2} \leq \cdots \leq a_{\mathcal{I}_n}$. Then, the entire coordinate axis of u can be divided into $n+2$ intervals, according to these $n+1$ end points of $\mathbf{a}$. Thus, when u is in each interval, the gradient can be analyzed as follows:

(i) if $u \in (-\infty, a_{\mathcal{I}_0})$, $\nabla J_1(u) = -\sum_{i=0}^{n} |b_i|$.

(ii) if $u \in (a_{\mathcal{I}_n}, +\infty)$, $\nabla J_1(u) = \sum_{i=0}^{n} |b_i|$.

(iii) if $u \in (a_{\mathcal{I}_k}, a_{\mathcal{I}_{k+1}})$, $0 < k < n$, $\nabla J_1(u) = \sum_{i=0}^{k} |b_{\mathcal{I}_i}| - \sum_{j=k+1}^{n} \left| b_{\mathcal{I}_j} \right|$, which is in-between the result of (i) and (ii).

(iv) given $u_1 \in (a_{\mathcal{I}_k}, a_{\mathcal{I}_{k+1}})$ and $u_2 \in (a_{\mathcal{I}_k+1}, a_{\mathcal{I}_{k+2}})$, the following inequality can be obtained according to (iii): $\nabla J_1(u_2) - \nabla J_1(u_1) = 2|b_{\mathcal{I}_{k+1}}| > 0$.

Thus it can be seen when u passes through all the intervals $(-\infty, a_{\mathcal{I}_0})$, $(a_{\mathcal{I}_0}, a_{\mathcal{I}_1})$, $\cdots$, $(a_{\mathcal{I}_{n-1}}, a_{\mathcal{I}_n})$, $(a_{\mathcal{I}_n}, +\infty)$ sequentially, $\nabla J_1(u)$ is increasing monotonously from negative to positive according to (i)-(iv). That is to say, (i)-(iv) proves the convexity of problem (5.55). As a result, the global minimum of $J_1(u)$ can be achieved in certain interval, which has the smallest absolute value of gradient.

Based on the above analysis of the gradient and the convexity, we have the following theorem to solve $J_1(u)$.

Theorem 5.4.1. *The optimal solution of $J_1(u)$ is given by,*

$$u^* = \operatorname*{argmin}_{u \in \mathcal{S}} J_1(u), \tag{5.59}$$

where $\mathcal{S} = \{0, \widetilde{x}_1/v_1, \widetilde{x}_2/v_2, \cdots, \widetilde{x}_n/v_n\}$.

Proof of Theorem 5.4.1. First, all the endpoints in the set $\mathcal{S}$ are sorted in an ascending order $\mathcal{I}$. Given any interval $(\mathcal{S}_{\mathcal{I}_k}, \mathcal{S}_{\mathcal{I}_{k+1}})$, we have $\nabla J_1(u) = \sum_{i=0}^{k} |b_{\mathcal{I}_i}| - \sum_{j=k+1}^{n} |b_{\mathcal{I}_j}|$. In this interval, $J_1(u)$ is a linear function, since $\nabla J_1(u)$ is a constant. Thus, the local minimum of $J_1(u)$ can be obtained by either $\mathcal{S}_{\mathcal{I}_k}$ or $\mathcal{S}_{\mathcal{I}_{k+1}}$, which one of them is decided by the sign of $\nabla J_1(u)$.

Obviously, $J_1(u)$ is a piecewise-linear function, when u passes through all the intervals along entire coordinate axis of u, i.e., $(-\infty, \mathcal{S}_{\mathcal{I}_0})$, $(\mathcal{S}_{\mathcal{I}_0}, \mathcal{S}_{\mathcal{I}_1})$, $\cdots$, $\big(\mathcal{S}_{\mathcal{I}_{n-1}}, \mathcal{S}_{\mathcal{I}_n}\big)$, $(\mathcal{S}_{\mathcal{I}_n}, +\infty)$. When u approaches to $\pm\infty$, $J_1(u)$ goes to infinity. Thus, the global minimum of the objective function $J_1(u)$ is obtained from one of endpoints in the set $\mathcal{S}$, which completes the proof. □

5.4.2.2 Solving for $J_{21}(u)$

The objective function $J_{21}(u)$ defined in Eq. (5.56) is rewritten equivalently as

$$\begin{aligned} J_{21}(u) &= \left\|\widetilde{\mathbf{x}} - u \cdot \mathbf{v}\right\|_1 + \lambda\sqrt{u^2 + c} \\ &= \sum_{i=1}^{n} |v_i| \left|\tfrac{\widetilde{x}_i}{v_i} - u\right| + \lambda\sqrt{u^2 + c} \\ &= \sum_{i=0}^{n} |b_i| \, |a_i - u| + \lambda\sqrt{u^2 + c}, \end{aligned} \tag{5.60}$$

where $\mathbf{a} = (0, \widetilde{x}_1/v_1, \cdots, \widetilde{x}_n/v_n)^T$, $\mathbf{b} = (0, v_1, \cdots, v_n)^T$.

Thus, the gradient of $J_{21}(u)$ with respect to u is computed as

$$\nabla J_{21}(u) = \sum_{i=0}^{n} |b_i| \cdot \text{sign}\,(u - a_i) + \lambda \frac{u}{\sqrt{u^2 + c}}. \tag{5.61}$$

As compared to $\nabla J_1(u)$, it becomes more difficult to analyze $\nabla J_{21}(u)$, since $\nabla\sqrt{u^2 + c}$ varies in certain interval.

Following the same procedure of solving $J_1(u)$, we still divide the entire coordinate axis of u into $n + 2$ intervals, using these $n + 1$ end points from the ascending order $\mathcal{I}$ of $\mathbf{a}$. Due to the reason that $\nabla\sqrt{u^2 + c}$ turns from negative to positive when

u crosses 0, 0 still can be treated as one of the endpoints. If $u \in (-\infty, a_{\mathcal{I}_0})$, $\nabla J_{21}(u)$ is negative, and can go to negative infinity when $u \to -\infty$. If $u \in (a_{\mathcal{I}_n}, +\infty)$, $\nabla J_{21}(u)$ is positive, and goes to positive infinity when $u \to +\infty$.

However, the variation tendency of $\nabla J_{21}(u)$ is not very obvious, when $u \in (a_{\mathcal{I}_k}, a_{\mathcal{I}_{k+1}})$. On the contrary, it is more easier to make a comparison between the gradients in any two adjacent intervals.

Given $u_1 \in (a_{\mathcal{I}_k}, a_{\mathcal{I}_{k+1}})$ and $u_2 \in (a_{\mathcal{I}_k+1}, a_{\mathcal{I}_{k+2}})$, we have the following inequality for the first term of $J_{21}(u)$

$$\nabla \|\widetilde{\mathbf{x}} - u_1 \cdot \mathbf{v}\|_1 \leq \nabla \|\widetilde{\mathbf{x}} - u_2 \cdot \mathbf{v}\|_1, \tag{5.62}$$

according to the gradient analysis of $J_1(u)$ when $u_1 \leq u_2$. Besides, we can establish the following inequality for the second term of $J_{21}(u)$

$$\nabla \sqrt{(u_1)^2 + c} \leq \nabla \sqrt{(u_2)^2 + c}, \tag{5.63}$$

since $\nabla \sqrt{(u)^2 + c}$ is a monotonously increasing function, which is based on the fact that $\nabla^2 \sqrt{(u)^2 + c} = c / \left(u^2 + c\right)^{2.5} > 0$.

Combining the results in Eqs. (5.62-5.63) together, we have the following equality for the gradient of $J_{21}(u)$

$$\nabla \|\widetilde{\mathbf{x}} - u_1 \cdot \mathbf{v}\|_1 + \lambda \cdot \nabla \sqrt{(u_1)^2 + c} \leq \nabla \|\widetilde{\mathbf{x}} - u_2 \cdot \mathbf{v}\|_1 + \lambda \cdot \nabla \sqrt{(u_2)^2 + c}. \tag{5.64}$$

That is to say, $\nabla J_{21}(u_1) \leq \nabla J_{21}(u_2)$.

On the other hand, when u approaches to $\pm\infty$, the gradient of $J_{21}(u_1)$ approaches to $\pm\infty$ respectively.

As it can be seen when u passes through all the intervals, $\nabla J_{21}(u)$ is increasing monotonously from negative to positive. Thus, the global minimum of the objective function $J_{21}(u)$ can be obtained in certain interval with the smallest absolute value of gradient, which also proves the convexity of $J_{21}(u)$.

Based on the above analysis of the gradient and the convexity, we have the following theorem to solve $J_{21}(u)$.

Theorem 5.4.2. *Let* $\widetilde{u} = \operatorname*{argmin}_{u \in \mathcal{S}} J_{21}(u)$*, where* $\mathcal{S} = \{0, \widetilde{x}_1/v_1, \cdots, \widetilde{x}_n/v_n\}$*. The optimal solution of* $J_{21}(u)$ *is given by*

$$u^* = \begin{cases} \widetilde{u}, \text{ if } \nabla J_{21}(\widetilde{u} - \epsilon) < 0, \ \nabla J_{21}(\widetilde{u} + \epsilon) > 0 \\ \operatorname{sign}\left(-d(\widetilde{u} + \epsilon)\right) \frac{c|d(\widetilde{u}+\epsilon)|}{\sqrt{\lambda^2 - d^2(\widetilde{u}+\epsilon)}}, \text{ if } \nabla J_{21}(\widetilde{u} \pm \epsilon) < 0 \\ \operatorname{sign}\left(-d(\widetilde{u} - \epsilon)\right) \frac{c|d(\widetilde{u}-\epsilon)|}{\sqrt{\lambda^2 - d^2(\widetilde{u}-\epsilon)}}, \text{ if } \nabla J_{21}(\widetilde{u} \pm \epsilon) > 0 \end{cases} \tag{5.65}$$

where $d(u) = \nabla \|\widetilde{\mathbf{x}} - u \cdot \mathbf{v}\|_1$*, and* ϵ *is a small positive number.*

Proof of Theorem 5.4.2. Since $\widetilde{u}$ achieves the minimum of $J_{21}(u)$ among these endpoints in the set $\mathcal{S}$, $\widetilde{u}$ should be either the left endpoint or the right endpoint in certain interval, where $|\nabla J_{21}(u)|$ is the smallest.

Suppose the optimal interval is $\left(\mathcal{S}_{\mathcal{I}_k}, \mathcal{S}_{\mathcal{I}_{k+1}}\right)$. If $\nabla J_{21}(\widetilde{u} \pm \epsilon) < 0$, $\widetilde{u}$ should be the left endpoint $\mathcal{S}_{\mathcal{I}_k}$. If $\nabla J_{21}(\widetilde{u} \pm \epsilon) > 0$, $\widetilde{u}$ should be the right endpoint $\mathcal{S}_{\mathcal{I}_{k+1}}$.

However, we will not have any optimal interval, if $\nabla J_{21}(u)$ could not approach to zero. That is to say, the optimal solution u^* should be the endpoint $\widetilde{u}$, where $\nabla J_{21}(\widetilde{u} - \epsilon) < 0$ and $\nabla J_{21}(\widetilde{u} + \epsilon) > 0$. In this case, we will have $|d(\widetilde{u})| > \lambda$, since $\nabla J_{21}(\widetilde{u}) \neq 0$ and $-1 < \nabla\sqrt{\widetilde{u}^2 + c} < 1$.

On the other hand, when $|d(u)| < \lambda$, we have $\nabla J_{21}(u) = 0$ in the optimal interval $\left(\mathcal{S}_{\mathcal{I}_k}, \mathcal{S}_{\mathcal{I}_{k+1}}\right)$. If $\nabla J_{21}(\widetilde{u} \pm \epsilon) < 0$, $\widetilde{u} = \mathcal{S}_{\mathcal{I}_k}$ and $d(\widetilde{u} + \epsilon) = d(u^*)$, since $d(\cdot)$ is a constant in the corresponding interval. By setting $\nabla J_{21}(u^*) = 0$, we have the following equality

$$d(\widetilde{u}+\epsilon) + \lambda \frac{u^*}{\sqrt{(u^*)^2 + c}} = 0, \tag{5.66}$$

which gives the optimal solution u^* as

$$u^* = \operatorname{sign}\left(-d(\widetilde{u} + \epsilon)\right) \frac{c|d(\widetilde{u} + \epsilon)|}{\sqrt{\lambda^2 - d^2(\widetilde{u} + \epsilon)}}. \tag{5.67}$$

If $\nabla J_{21}(\widetilde{u} \pm \epsilon) > 0$, $\widetilde{u} = \mathcal{S}_{\mathcal{I}_{k+1}}$ and $d(\widetilde{u} - \epsilon) = d(u^*)$. Then, the optimal solution u^* is obtained in the similar way as

$$u^* = \operatorname{sign}\left(-d(\widetilde{u} - \epsilon)\right) \frac{c|d(\widetilde{u} - \epsilon)|}{\sqrt{\lambda^2 - d^2(\widetilde{u} - \epsilon)}}. \tag{5.68}$$

Based on the above analysis of three cases, we have the optimal solution u^* of $J_{21}(u)$ defined in Eq. (5.65), which completes the proof. □

5.4.3 Implementation Details of Exact Solver Based Optimization Algorithm

As it can be seen in Eq. (5.59) and Eq. (5.65), the exact solver of $J_1(u)$ and $J_{21}(u)$ is very efficient that the optimal solution u^* can be obtained in linear time.

To minimize L_1-/L_{21}-norm penalty based robust L_1-PCA models, we apply exact solver to obtain the optimal solution with respect to a single entry of $\mathbf{U}$ and $\mathbf{V}$ at each time. Once all the elements of $\mathbf{U}$ are updated, $\mathbf{V}$ will be updated element-by-element in the following. Thus, we can minimize $\mathbf{U}$ and $\mathbf{V}$ alternatively until the objective function converges.

In the previous section, we only discussed the case that exact solver is applied to update the elements of matrix $\mathbf{U}$, i.e., principal directions. To update the elements of matrix $\mathbf{V}$, i.e., principal components, we can reuse these exact solvers via transposing the L_1-norm based loss term.

After transpose, $\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_1$ becomes to $\|\mathbf{Y} - \mathbf{V}\mathbf{U}^T\|_1$, where $\mathbf{Y} = \mathbf{X}^T$. Thus, matrix $\mathbf{V}$ can be updated using same exact solvers with following replacements such as

$$u \leftarrow v, \mathbf{v} \leftarrow \mathbf{u}, \widetilde{\mathbf{x}} \leftarrow \widetilde{\mathbf{y}},$$

where v is the (j_0, h_0)-th entry of the corresponding matrix $\mathbf{V}$ ($j_0 = 1, \cdots, n$, $h_0 = 1, \cdots, k$), $\mathbf{u}$ is the h_0-th column of the corresponding matrix $\mathbf{U}$, $\widetilde{\mathbf{y}}$ is the transpose of the j_0-th row of the corresponding matrix $\widetilde{\mathbf{Y}}$.

Algorithm 5 UPDATE-L1($\mathbf{X}$, $\mathbf{U}$, $\mathbf{V}$, d, k, λ)

```
Input: $\mathbf{X}$, $\mathbf{U}$, $\mathbf{V}$, $d$, $k$, $\lambda$.
Output: $\mathbf{U}$.
for $i_0 = 1$ to $d$ do
    for $h_0 = 1$ to $k$ do
        $\widetilde{\mathbf{X}} = \mathbf{X} - \mathbf{U}\mathbf{V}^T + \mathbf{U}_{h_0}(\mathbf{V}_{h_0})^T$.
        $\widetilde{\mathbf{x}} = (\widetilde{\mathbf{X}}^{i_0})^T$.
        $\mathbf{v} = \mathbf{V}_{h_0}$.
        $\mathbf{U}_{i_0 h_0} = \underset{u}{\operatorname{argmin}}\, J_1(u, \widetilde{\mathbf{x}}, \mathbf{v}, \lambda)$ via Eq. (5.59).
    end for
end for
return $\mathbf{U}^*$.
```

Algorithm 6 UPDATE-L21($\mathbf{X}$, $\mathbf{U}$, $\mathbf{V}$, d, k, λ)

```
Input: $\mathbf{X}$, $\mathbf{U}$, $\mathbf{V}$, $d$, $k$, $\lambda$.
Output: $\mathbf{U}$.
for $i_0 = 1$ to $d$ do
    for $h_0 = 1$ to $k$ do
        $\widetilde{\mathbf{X}} = \mathbf{X} - \mathbf{U}\mathbf{V}^T + \mathbf{U}_{h_0}(\mathbf{V}_{h_0})^T$.
        $c = \|\mathbf{U}_{h_0}\|_2^2 - \mathbf{U}_{i_0 h_0}^2$.
        $\widetilde{\mathbf{x}} = (\widetilde{\mathbf{X}}^{i_0})^T$.
        $\mathbf{v} = \mathbf{V}_{h_0}$.
        $\mathbf{U}_{i_0 h_0} = \underset{u}{\operatorname{argmin}}\, J_{21}(u, \widetilde{\mathbf{x}}, \mathbf{v}, c, \lambda)$ via Eq. (5.65).
    end for
end for
return $\mathbf{U}^*$.
```

Algorithm 7 Exact Solver Based Optimization Algorithm for Solving L_1-/L_{21}-norm Penalty Based Robust L_1-PCA Model.

```
Input: data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$, rank $k$, hyperparameter $\lambda$.
Output: principal directions $\mathbf{U} \in \mathbb{R}^{d \times k}$, principal components $\mathbf{V} \in \mathbb{R}^{n \times k}$.
Parameter settings: $t = 0$, $\epsilon = 1e - 6$, $\mathbf{Y} = \mathbf{X}^T$.
Initialization:
(i) SVD initialization: $\mathbf{U}^{(t)}, \mathbf{V}^{(t)} = \text{svd}(\mathbf{X}, k)$;
(ii) Random initialization: $\mathbf{U}^{(t)} = \text{rand}(p, k)$, $\mathbf{V}^{(t)} = \text{rand}(n, k)$.
repeat
    (i) For $L_1$-norm penalty, $\mathbf{U}^{(t+1)} = \text{UPDATE-L1}(\mathbf{X}, \mathbf{U}^{(t)}, \mathbf{V}^{(t)}, d, k, \lambda)$;
    (ii) For $L_{21}$-norm penalty, $\mathbf{U}^{(t+1)} = \text{UPDATE-L21}(\mathbf{X}, \mathbf{U}^{(t)}, \mathbf{V}^{(t)}, d, k, \lambda)$.
    (i) For $L_1$-norm penalty, $\mathbf{V}^{(t+1)} = \text{UPDATE-L1}(\mathbf{Y}, \mathbf{V}^{(t)}, \mathbf{U}^{(t+1)}, n, k, \lambda)$;
    (ii) For $L_{21}$-norm penalty, $\mathbf{V}^{(t+1)} = \text{UPDATE-L21}(\mathbf{Y}, \mathbf{V}^{(t)}, \mathbf{U}^{(t+1)}, n, k, \lambda)$.
    $t = t + 1$.
until Objective function converges, i.e., $|J_t - J_{t-1}|/J_t < \epsilon$.
return Optimal solutions: $\mathbf{U}^*$, $\mathbf{V}^*$.
```

The exact solver based optimization algorithm for solving L_1-/L_{21}-norm penalty based robust L_1-PCA model is summarized in Algorithms 5-7.

First, we introduce the subroutine of exact solver for updating matrix $\mathbf{U}$, where Algorithm 5 solves the L_1-norm penalty via Theorem 5.4.1 and Algorithm 6 solves the L_{21}-norm penalty via Theorem 5.4.2. On the other hand, we can update matrix $\mathbf{V}$ using the same subroutine, via transposing the L_1-norm based loss term.

The completed framework of optimizing robust L_1-PCA models is given in Algorithm 7. We also initialize $\mathbf{U}$ and $\mathbf{V}$ via SVD initialization or random initialization, which is already explained in ALM based optimization algorithm. Then, at each iteration, $\mathbf{U}$ and $\mathbf{V}$ are updated alternatively via exact solvers, see Algorithms 5-6. Finally, the optimization algorithm coverges until $|J_t - J_{t-1}|/J_t$ is smaller than the threshold ϵ.

5.5 Experiments

5.5.1 Benchmark Dataset

Extensive experiments on benchmark dataset are performed to evaluate the effectiveness of the proposed robust model and the optimization algorithm.

In the benchmark dataset, four hundreds face images are collected by AT&T Laboratories Cambridge, including ten different images of each of forty distinct persons. Each face image was taken under varying lighting and facial expression.

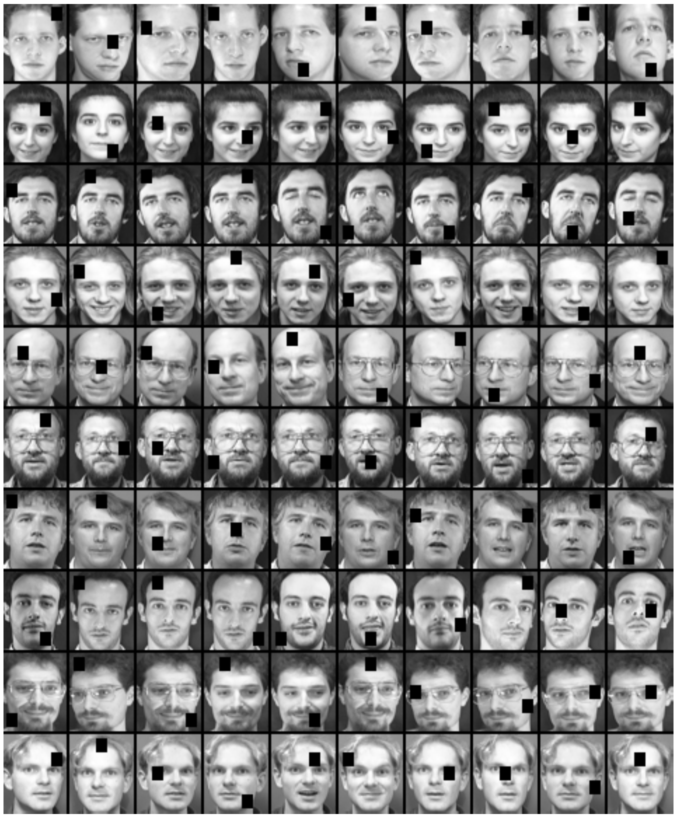


Figure 5.1: Corrupted face images of AT&T dataset.

In the experiments, the original face image is resized to a lower ratio 56 × 46. Each face image is represented by a 2576-dimensional vector. Towards verifying the robustness of the proposed data reconstruction model, we add rectangular noises at a random position in each face image.

The detail of the corrupted face image is shown in Figure 5.1, where the rectangular corruption is filled with black pixels.

5.5.2 Result and Analysis

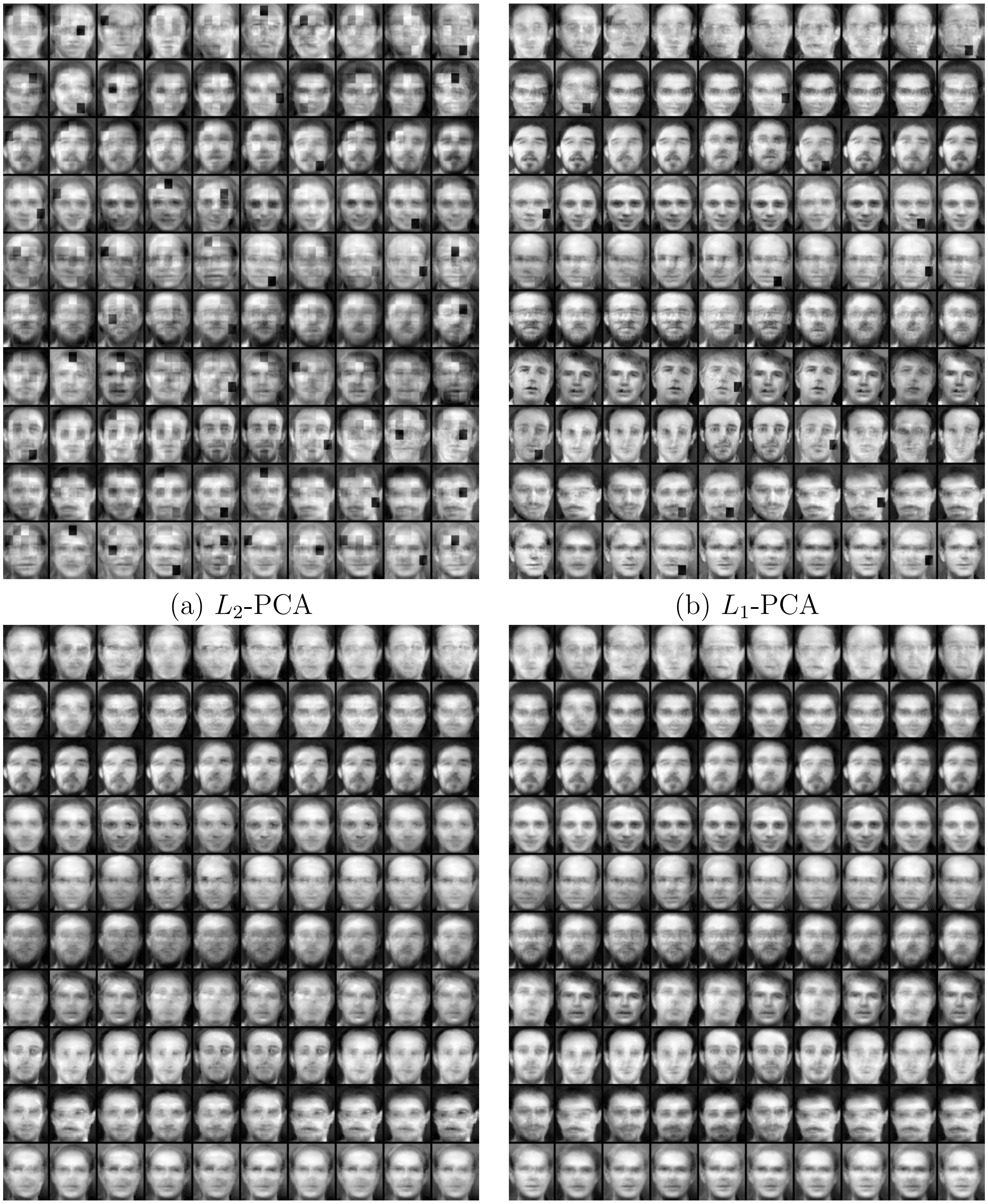
In the following, we apply robust PCA based low-rank and sparse data reconstruction method to reconstruct corrupted face images. The effectiveness/robustness of the proposed model is evaluated based on the quality of reconstructed results.

5.5.2.1 Robustness of L_1-/L_{21}-norm Penalty

With assumption of the underlying noise in the principal directions and principal components, we derive the L_1-/L_{21}-norm penalty based robust L_1-PCA models. Our derivation mathematically proves the connection between robustness and regularization. To further verify this fact, we apply robust L_1-PCA models to reconstruct the corrupted face images.

In the experiments, we apply data reconstruction methods (including L_2-PCA, L_1-PCA, L_1-norm penalty based robust L_1-PCA, and L_{21}-norm penalty based robust L_1-PCA) to reconstructed the corrupted face images shown in Figure 5.1. The rank parameter k is set as 15. The result of L_2-PCA (i.e. the SVD result of the corrupted input data) is used as the initialization for other three methods. The reconstructed results are shown in Figure 5.2.

As it can be seen in Figure 5.2-(a), L_2-PCA completely fails to reconstruct the corrupted input data, since L_2 is not robust enough when dealing with a large number



(a) L_2-PCA

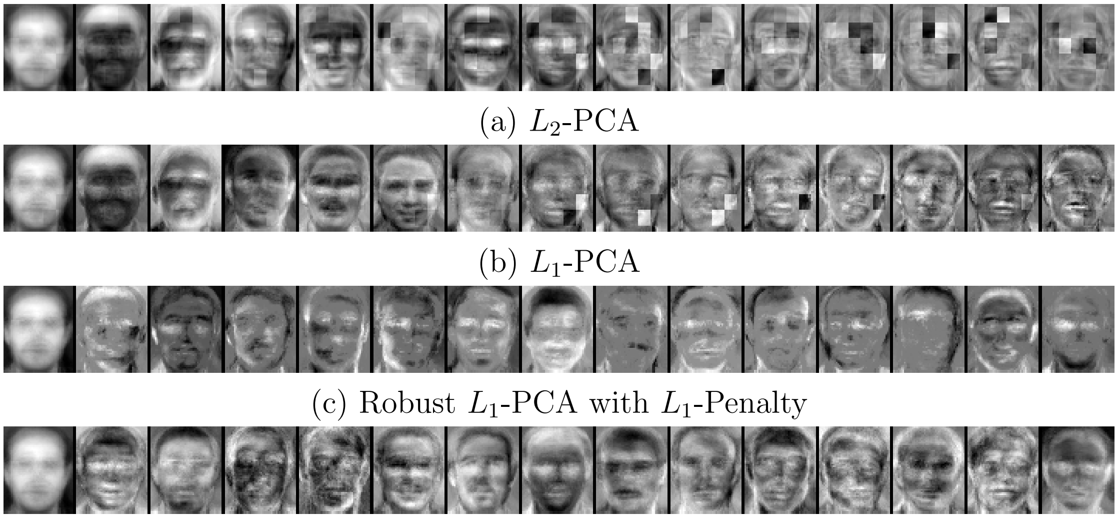
(b) L_1-PCA

(c) Robust L_1-PCA with L_1-Penalty

(d) Robust L_1-PCA with L_{21}-Penalty

Figure 5.2: Robust L_1-PCA versus state-of-the-arts. Reconstructed results from four different methods.

of noises. On the contrary, L_1-PCA is capable of removing most of the rectangular noises, see Figure 5.2-(b), meanwhile recovering the original faces with a good quality. Our proposed robust L_1-PCA models can further improve the robustness of the reconstructed results as compared to L_1-PCA, see Figure 5.2-(c,d), where the remaining noises shown in Figure 5.2-(b) are all removed out.



(a) L_2-PCA

(b) L_1-PCA

(c) Robust L_1-PCA with L_1-Penalty

(d) Robust L_1-PCA with L_{21}-Penalty

Figure 5.3: Robust L_1-PCA versus state-of-the-arts. Learned principal directions from four different methods.

On the other hand, learned principal directions are shown in Figure 5.3. Both L_2-PCA and L_1-PCA learn noisy principal directions, see Figure 5.3-(a,b), since L_2-PCA models small noises and L_1-PCA models large noises in original feature space. However, L_1-PCA with L_1-/L_{21}-penalty learns clean principal directions, see Figure 5.3-(c,d), because large noises are modeled in latent feature space. Thus, L_1-/L_{21}-penalty terms can help L_1-PCA model to remove the noises in principal directions.

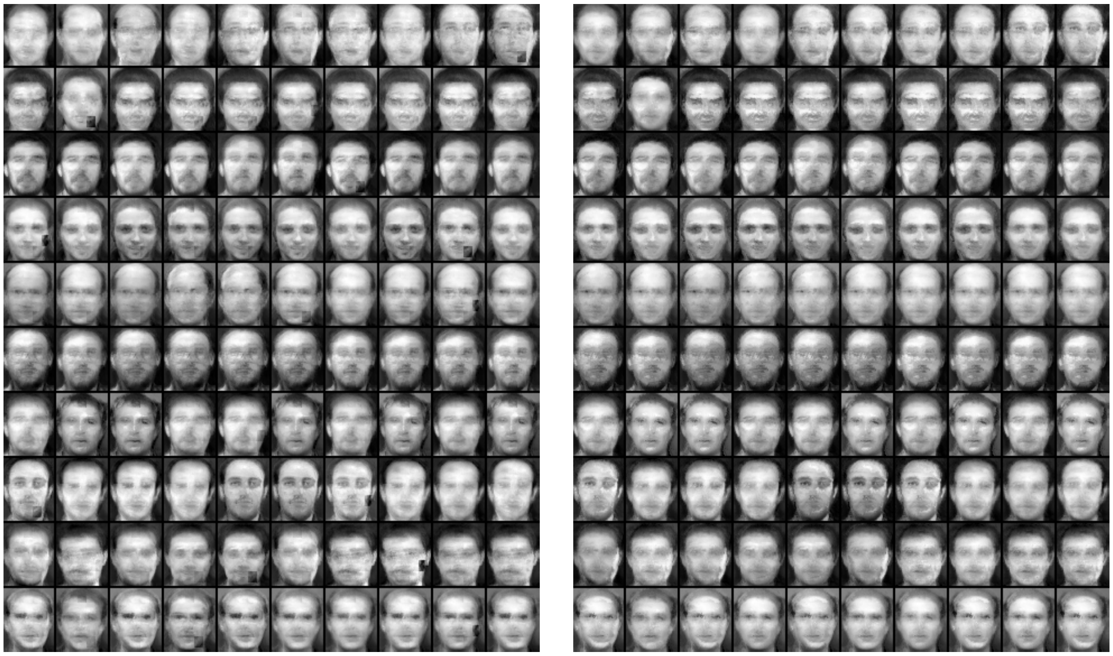
As a conclusion, L_1-PCA with L_1-/L_{21}-penalty achieves the best result among these methods. Thus, the fact that the regularization helps improving the robustness of machine learning models is verified.

5.5.2.2 Effectiveness of Exact Solver

In the following, we conduct experiments on the corrupted face images to verify the effectiveness of exact solver based optimization algorithm, as compared to ALM based optimization algorithm.

Details of the comparison are explained as follows. First, principal directions $\mathbf{U}^{(0)}$ and principal components $\mathbf{V}^{(0)}$ are initialized by the results of SVD. Starting from $\mathbf{U}^{(0)}(\mathbf{V}^{(0)})^T$, ALM based optimization algorithm reconstructs the corrupted face image as $\mathbf{U}^{(1)}(\mathbf{V}^{(1)})^T$. To further improve the reconstructed result $\mathbf{U}^{(1)}(\mathbf{V}^{(1)})^T$, exact solver based optimization algorithm is applied to obtain the final reconstructed result as $\mathbf{U}^{(2)}(\mathbf{V}^{(2)})^T$.

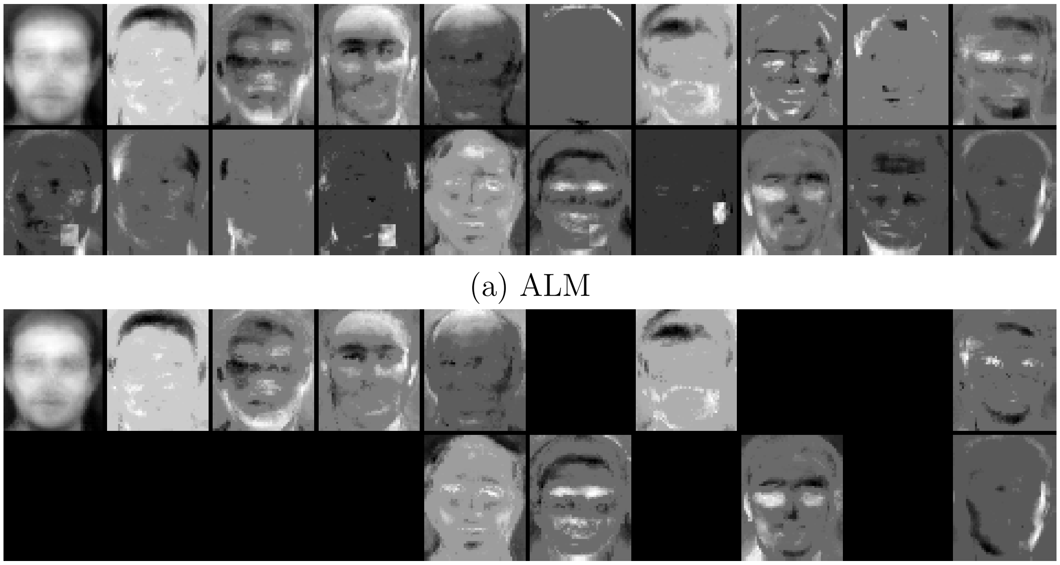
First of all, we apply optimization algorithms to solve the L_1-norm penalty based robust L_1-PCA model. Here, we use the first 10 person in AT&T dataset as the input matrix $\mathbf{X} \in \mathbb{R}^{2576\times 100}$ ($d = 2576$, $n = 100$), and set the rank k to 20. The comparison results between ALM and exact solver are shown in Figures 5.4-5.5. In Figure 5.4-(a), ALM based optimization algorithm removes most of the corruptions (see Figure 5.1) in face images. In Figure 5.4-(b), exact solver based optimization algorithm removes all the remaining rectangular noises shown in the ALM results, for example 10th face image in 1st row and 2nd face image in 2nd row. Thus it can be seen that exact solver obtains a better reconstructed result than ALM. Additionally, ALM learns the noisy principal directions such as 1st and 4th face images in 2nd row, see Figure 5.5-(a), that's why the reconstructed result still has a few corruptions. On the contrary, exact solver learns the clean principal directions, see Figure 5.5-(b), and recovers the original face images with a good quality. As a result, exact solver improves the robustness of ALM in solving the L_1-norm penalty based robust L_1-PCA model, which verifies the effectiveness of exact solver based optimization algorithm.



(a) ALM

(b) Exact Solver

Figure 5.4: Exact Solver versus ALM. Reconstructed results of L_1-norm penalty based robust L_1-PCA model from two optimization algorithms.

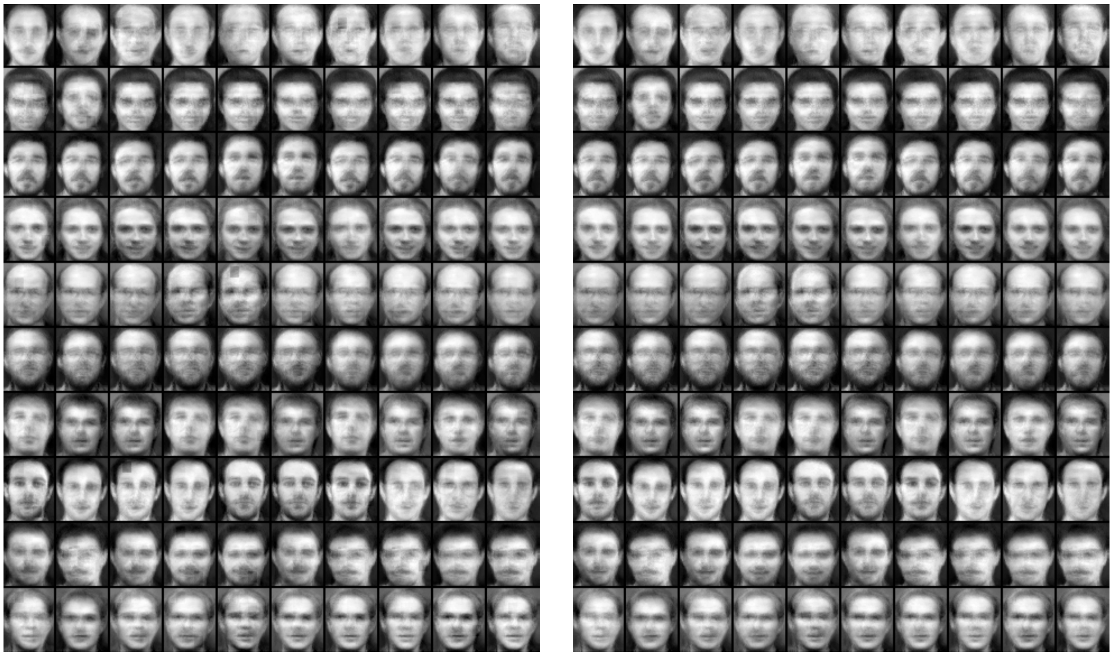


(a) ALM

(b) Exact Solver

Figure 5.5: Exact Solver versus ALM. Learned principal directions of L_1-norm penalty based robust L_1-PCA model from two optimization algorithms.

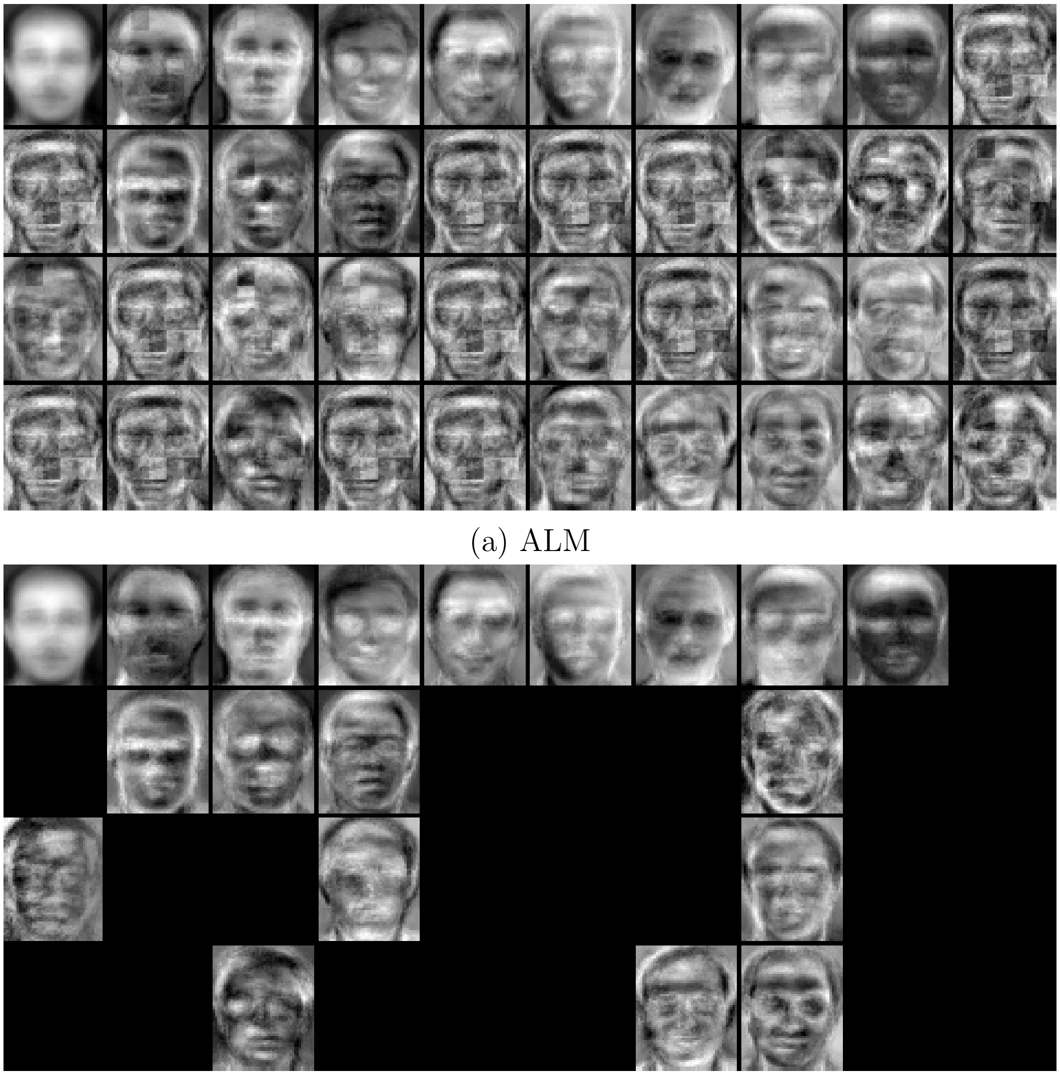
Secondly, we apply optimization algorithms to solve the L_{21}-norm penalty based robust L_1-PCA model. Here, we use the first 20 person in AT&T dataset as the input matrix $\mathbf{X} \in \mathbb{R}^{2576\times 200}$ ($d = 2576$, $n = 200$), and set the rank k as 40. The comparison results between ALM and exact solver are shown in Figures 5.6-5.7, where Figure 5.6 shows the reconstructed results and Figure 5.7 shows the learned principal directions. On the one hand, exact solver obtains better reconstructed results as compared to ALM, see Figure 5.6, for example 5th face image in 5th row, 3rd face image in 8th row, and 5th face in 9th row. On the other hand, exact solver learns the clean principal directions in Figure 5.7-(b), while ALM learns the noisy principal directions in Figure 5.7-(a). As a result, exact solver improves the robustness of ALM in solving the L_{21}-norm penalty based robust L_1-PCA model, which also verifies the effectiveness of exact solver based optimization algorithm.



(a) ALM

(b) Exact Solver

Figure 5.6: Exact Solver versus ALM. Reconstructed results of L_{21}-norm penalty based robust L_1-PCA model from two optimization algorithms.



(a) ALM

(b) Exact Solver

Figure 5.7: Exact Solver versus ALM. Learned principal directions of L_{21}-norm penalty based robust L_1-PCA model from two optimization algorithms.

5.6 Conclusion

In this work, we introduce robust PCA based low-rank and sparse data reconstruction method, which models the underlying noises in the lower-dimensional latent feature space. For purpose of simplifying the corresponding multivariate optimization problem, we derive the tight upper bounds of robust L_1-PCA models, i.e., the L_1-/L_{21}-norm penalty based robust L_1-PCA model. At the same time, our derivation theoretically proves the connection between the robustness and the regularization from a robust point of view.

Then, an efficient augmented Lagrangian multiplier based optimization algorithm is proposed to minimize the derived tight upper bounds, where the original problem is decomposed into several subproblems with close-form solutions. To further improve the robustness of the reconstructed results, we present an exact solver based optimization algorithm. Exact solver minimizes the L_1-/L_{21}-norm penalty based robust L_1-PCA models with respect to a single entry of principal directions and principal components at each time. Because of resolving this simplified optimization problem in linear time, exact solver works efficiently with alternative optimization technique.

Experimental results on AT&T datasets show that L_1-/L_{21}-norm penalty based robust L_1-PCA models obtain better reconstructed results of the corrupted face images than other methods, such as L_2-PCA and L_1-PCA. On the other hand, exact solver based optimization can further improves the robustness of reconstructed results, as compared to augmented Lagrangian multiplier based optimization algorithm. Thus it can be seen that experimental results verify the effectiveness of the proposed robust data reconstruction model and the proposed optimization algorithm.

CHAPTER 6

CONCLUSION

Feature selection and data reconstruction are crucial to data analysis, since it helps users to build better machine learning models for real-world applications such as classification, clustering, etc. As a result, we focus on developing several robust and flexible learning models to improve the efficiency and the effectiveness of feature selection and data reconstruction, which can pave the way for data modeling, that is the key step of data analysis.

First, we derive LASSO from a probabilistic point of view. The proposed probabilistic selective ridge regression further explains the sparsity of ℓ_1-norm, and introduces a new ranking method to measure the importance of features. Based on probabilistic LASSO, we extend this two-class method to solve multi-class problem, where we are adding a probabilistic selection vector for each class separately. Thus, we apply this probability-derived $\ell_{1,2}$-norm to select discriminative features for each class, so as to provide certain flexibility in feature selection. An auxiliary function is introduced to iteratively optimize $\ell_{1,2}$-norm regularized linear regression problem, with vigorous convergence guarantee. Empirical studies show that our $\ell_{1,2}$-norm based flexible feature selection resolves the inflexibility of class-shared feature selection such as the widely used $\ell_{2,1}$-norm, and further improves the performance on multi-class classification.

Additionally, we propose a novel "exclusive $\ell_{2,1}$" regularization (short for $\ell_{2,1}$ with exclusive lasso) to select robust and flexible features, which synergistically combines the advantages of different sparsity-induced norms. Exclusive $\ell_{2,1}$ regularization not only increases the robustness via $\ell_{2,1}$-norm but also provides the flexibility via $\ell_{1,2}$-norm. To resolve the inefficiency of existing solvers such as re-weighted method and coordinate descent method, we propose a sorting-based explicit approach to solve $\ell_{1,2}$-norm based proximal operator-type problem. Besides, we also point out some interesting property of $\|\mathbf{w}\|_1^2$ regularization as compared to $\|\mathbf{w}\|_1$ regularization, which helps us to have a better understanding of so-called "exclusive sparsity" of $\ell_{1,2}$-norm. Finally, an augmented Lagrangian multiplier based optimization method is presented to iteratively solve the exclusive $\ell_{2,1}$ regularization in a row-wise fashion, which greatly reduces the computational cost and is well-suited for large-scale data. Empirical studies show that the proposed optimization algorithm converges fast and is very efficient in real-world applications; the performance of exclusive $\ell_{2,1}$ regularization is not sensitive to the change of hyperparameters; and the proposed exclusive $\ell_{2,1}$ regularization achieves higher classification accuracy on multi-class problems as compared to state-of-the-arts, such as $\ell_{2,1}$ and exclusive lasso.

In this thesis, we also present deep ℓ_1-autoencoder networks for data reconstruction. As compared to linear methods using nuclear norm and other low-ranks models, our proposed deep robust data reconstruction model is more capable of capturing the complicated intrinsic property of the corrupted input data, because of multi-layer and non-linear architectures. To resolve the black spot problem incurred by ℓ_1 loss based networks using ReLU activation, we further introduce a smoothed version of ReLU (sReLU), which can provide a small positive gradient when the input is smaller than 0. Thus, sReLU has the strength to drive black spots in the reconstructed output up from zero. As a result, the proposed ℓ_1-autoencoder networks using sReLU activation will not produce the black spot any more, and further improve the quality of reconstructed results as the number of layers is increasing. Empirical studies show that our proposed ℓ_1-autoencoder network is able to remove various kinds of occlusions in the input data, and achieves a lower noise-free reconstruction error as compared to state-of-the-arts such as robust PCA and L_1-PCA.

On the other hand, we propose a robust PCA based low-rank and sparse data reconstruction method to automatically remove the noise in the data. In the proposed robust L_1-PCA method, the underlying noises in principal directions and components are modeled by Laplacian distribution. Based on the robust L_1-PCA model, we derive two tight upper bounds of the original objective function, meanwhile which theoretically proves that the regularization improves the robustness of learning models. To optimize L_1-/L_{21}-norm penalty based robust L_1-PCA models, we first introduce an augmented Lagrangian multiplier based optimization algorithm. Then, "exact solver" algorithm is proposed to further improve the robustness of the data reconstruction. Empirical studies show that our proposed robust L_1-PCA models obtain better reconstructed results as compared to state-of-the-arts. Additionally, the proposed exact solver based optimization algorithm solves the multivariate problem more efficiently as compared to standard ALM based optimization algorithm, thus further improving the quality of the reconstruction.

In summary, we developed several robust and flexible learning models to conduct feature selection and data reconstruction from different point of views. It turns out that our proposed novel approaches not only successfully resolves the problems and limitations of existing methods, but also further improves the performance on real-world applications.

REFERENCES

[1] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, “A review of microarray datasets and applied feature selection methods,” *Information Sciences*, vol. 282, pp. 111–135, Oct. 2014.

[2] S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of discrimination methods for the classification of tumors using gene expression data,” *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.

[3] Y. Saeys, I. n. Inza, and P. Larrañaga, “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Sept. 2007.

[4] Y. Gao and G. Church, “Improving molecular cancer class discovery through sparse non-negative matrix factorization,” *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.

[5] D. Eigen, D. Krishnan, and R. Fergus, “Restoring an image taken through a window covered with dirt or rain,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 633–640.

[6] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, June 2005, pp. 60–65.

[7] S. V. Vaseghi, *Advanced Digital Signal Processing and Noise Reduction.* John Wiley & Sons, 2006.

[8] A. M. Martínez and A. C. Kak, “PCA versus LDA,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.

[9] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003.

[10] C. Ding and H. Peng, “Minimum redundancy feature selection from microarray gene expression data,” in *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*, Aug 2003, pp. 523–528.

[11] M. Robnik-Šikonja and I. Kononenko, “Theoretical and empirical analysis of relieff and rrelieff,” *Machine Learning*, vol. 53, no. 1, pp. 23–69, Oct 2003.

[12] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, Aug 2005.

[13] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1, pp. 389–422, Jan 2002.

[14] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[15] J. Zhu, S. Rosset, R. Tibshirani, and T. J. Hastie, “1-norm support vector machines,” in *Advances in Neural Information Processing Systems 16*, 2004, pp. 49–56.

[16] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Machine Learning*, vol. 73, no. 3, pp. 243–272, Dec 2008.

[17] J. Liu, S. Ji, and J. Ye, “Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 339–348.

[18] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 1813–1821.

[19] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan, "Feature selection based on structured sparsity: A comprehensive study," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1490–1507, July 2017.

[20] C. Ding, D. Zhou, X. He, and H. Zha, "R1-pca: Rotational invariant ℓ_1-norm principal component analysis for robust subspace factorization," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 281–288.

[21] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for $\ell_{1,\infty}$ regularization," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 857–864.

[22] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 272–279.

[23] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji, "Discriminant sparse neighborhood preserving embedding for face recognition," *Pattern Recognition*, vol. 45, no. 8, pp. 2884 – 2893, 2012.

[24] C.-Y. Lu, H. Min, J. Gui, L. Zhu, and Y.-K. Lei, "Face recognition via weighted sparse representation," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 111 – 116, 2013.

[25] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *The Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 12 2009.

[26] Y. Zhou, R. Jin, and S. Hoi, “Exclusive lasso for multi-task feature selection,” in *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2010, pp. 988–995.

[27] D. Kong, R. Fujimaki, J. Liu, F. Nie, and C. Ding, “Exclusive feature learning on arbitrary structures via $\ell_{1,2}$ -norm,” in *Advances in Neural Information Processing Systems 27*, 2014, pp. 1655–1663.

[28] F. Campbell and G. I. Allen, “Within group variable selection through the exclusive lasso,” *Electronic Journal of Statistics*, vol. 11, no. 2, pp. 4220–4257, 2017.

[29] L. Breiman, “Better subset regression using the nonnegative garrote,” *Technometrics*, vol. 37, no. 4, pp. 373–384, Nov. 1995.

[30] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 1999.

[31] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, pp. 788–91, 11 1999.

[32] C. H. Q. Ding, T. Li, and M. I. Jordan, “Convex and semi-nonnegative matrix factorizations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[33] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*, 2001, pp. 556–562.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[35] A. I. Su, J. B. Welsh, L. M. Sapinoso, S. G. Kern, *et al.*, “Molecular classification of human carcinomas by use of gene expression signatures,” *Cancer Research*, vol. 61, no. 20, pp. 7388–7393, 2001.

[36] K. Yang, Z. Cai, J. Li, and G. Lin, “A stable gene selection in microarray data analysis,” *BMC Bioinformatics*, vol. 7, no. 1, p. 228, Apr 2006.

[37] A. Bhattacharjee, W. G. Richards, J. Staunton, C. Li, *et al.*, “Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 24, pp. 13 790–13 795, 2001.

[38] E.-Y. Kwon, S.-K. Shin, Y.-Y. Cho, U. Ju Jung, *et al.*, “Time-course microarrays reveal early activation of the immune transcriptome and adipokine dysregulation leads to fibrosis in visceral adipose depots during diet-induced obesity,” *BMC genomics*, vol. 13, p. 450, 09 2012.

[39] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[40] L. Wang, J. Zhu, and H. Zou, “Hybrid huberized support vector machines for microarray classification,” in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 983–990.

[41] D. Ming, C. Ding, and F. Nie, “A probabilistic derivation of lasso and l12-norm feature selections,” in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, pp. 4586–4593.

[42] J. Cavazza, P. Morerio, B. Haeffele, C. Lane, V. Murino, and R. Vidal, “Dropout as a low-rank regularizer for matrix factorization,” in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, vol. 84, 09–11 Apr 2018, pp. 435–444.

[43] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[44] T. Sim, S. Baker, and M. Bsat, “The cmu pose, illumination, and expression (pie) database,” in *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, 2002, pp. 53–58.

[45] C. L. Nutt, D. R. Mani, R. A. Betensky, P. Tamayo, *et al.*, “Gene expression-based classification of malignant gliomas correlates better with survival than histological classification,” *Cancer Research*, vol. 63, no. 7, pp. 1602–1607, 2003.

[46] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, *et al.*, “Multiclass cancer diagnosis using tumor gene expression signatures,” *Proceedings of the National Academy of Sciences*, vol. 98, no. 26, pp. 15 149–15 154, 2001.

[47] P. M. Ciarelli and E. Oliveira, “Agglomeration and elimination of terms for dimensionality reduction,” in *2009 Ninth International Conference on Intelligent Systems Design and Applications*, Nov 2009, pp. 547–552.

[48] I. Jolliffe, *Principal component analysis.* New York: Springer Verlag, 2002.

[49] Q. Ke and T. Kanade, “Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming,” in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2005, pp. 739–746.

[50] M. Zhang and C. Ding, “Robust tucker tensor decomposition for effective image representation,” in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 2448–2455.

[51] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, 1st ed. Athena Scientific, 1996.

[52] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization,” in *Advances in Neural Information Processing Systems 22*, 2009, pp. 2080–2088.

[53] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[54] J.-F. Cai, E. J. Candès, and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[56] J. Xie, R. B. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *Proceedings of the 33nd International Conference on Machine Learning*, 2016, pp. 478–487.

[57] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[58] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[59] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, no. 2, 2014, pp. 1764–1772.

[60] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 3128–3137.

[61] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[62] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 1613–1622.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[64] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1096–1103.

[65] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, vol. 15, 2011, pp. 315–323.

[66] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 2074–2082.

[67] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Penksy, "Sparse convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 806–814.

[68] M. aurelio Ranzato, Y. lan Boureau, and Y. L. Cun, "Sparse feature learning for deep belief networks," in *Advances in Neural Information Processing Systems 20*, 2008, pp. 1185–1192.

[69] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 1135–1143.

[70] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 807–814.

[71] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia, "Incorporating second-order functional knowledge for better option pricing," in *Advances in Neural Information Processing Systems 13*, 2001, pp. 472–478.

[72] R. Salakhutdinov, A. Mnih, and G. E. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 791–798.

[73] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[74] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9, 13–15 May 2010, pp. 249–256.

[75] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, Aug 1989.

[76] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[77] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, Jan. 2001.

[78] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[79] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[80] D. Kong, C. Ding, and H. Huang, “Robust nonnegative matrix factorization using l21-norm,” in *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2011, pp. 673–682.

[81] D. Luo, C. Ding, and H. Huang, “Toward structural sparsity: an explicit l2/l0 approach,” *Knowledge and Information Systems*, vol. 36, no. 2, pp. 411–438, Aug 2013.

[82] D. Luo, C. Ding, F. Nie, and H. Huang, “Cauchy graph embedding,” in *Proceedings of the 28th International Conference on Machine Learning*, June 2011, pp. 553–560.

BIOGRAPHICAL STATEMENT

Di Ming was born in Zigong, Sichuan, China, in 1989. He started his Ph.D. study in Department of Computer Science and Engineering at University of Texas at Arlington since August, 2014, and defended in January, 2020. His thesis advisor is Dr. Chris Ding, and his primary research interest lies in the field of machine learning, deep learning, optimization, computer vision, medical image computing. Prior to that, he received his master degree in Computer Science from University of Electronic Science and Technology of China in June, 2014, and received his bachelor degree in Computer Science from Sichuan Agricultural University in June, 2011.