Towards Addressing Key Visual Processing Challenges in Social Media Computing

by

Xu Zhou

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2018 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Sharon Hsiao
Hasan Davulcu
Yezhou Yang

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Visual processing in social media platforms is a key step in gathering and understanding information in the era of Internet and big data. Online data is rich in content, but its processing faces many challenges including: varying scales for objects of interest, unreliable and/or missing labels, the inadequacy of single modal data and difficulty in analyzing high dimensional data. Towards facilitating the processing and understanding of online data, this dissertation primarily focuses on three challenges that I feel are of great practical importance: handling scale differences in computer vision tasks, such as facial component detection and face retrieval, developing efficient classifiers using partially labeled data and noisy data, and employing multi-modal models and feature selection to improve multi-view data analysis. For the first challenge, I propose a scale-insensitive algorithm to expedite and accurately detect facial landmarks. For the second challenge, I propose two algorithms that can be used to learn from partially labeled data and noisy data respectively. For the third challenge, I propose a new framework that incorporates feature selection modules into LDA models.

I dedicate this dissertation to Mr. Y.

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr Baoxin Li. He is a dedicated mentor. I won't be able to complete my Ph.D study without his help. His always gives encouragement and endless professional support whenever I need in my study. He spends a lot of time not only on helping me with specific problems but also on showing me a big picture of research on the field I am working on.

I would also like to thank Dr. Hasan Davulcu, Dr Sharon Hsiao and Dr. Yezhou Yang for their helpful advice on my research proposal and Ph.D. dissertation, and for sharing their precious time to serve as my dissertation committee members.

Furthermore, I would like to thank Daniel and Hakan who were my mentor and manager respectively when I worked as an intern. With their guidance, I learned a lot from my first industry experience and enjoy a meaningful summer.

My colleagues in Dr Baoxin Li's research group inspired me a lot through my entire Ph.D. study. Without their help and project cooperations, I would not be able to finish my dissertation. They are helpful and willing to share time and energy on my research problems and projects. I learned a lot from them through their important feedback. These are my previous and current team members and my friends: Dr. Qiang Zhang, Nan Li, Dr. Zheshen Wang, Dr. Peng Zhang, Dr. Devi Archana Paladugu, Dr. Zhigang Tu, Dr. Ke Bai, Jing Lu, Hima Bindu, Jun Cao, Dr.Parag Chandakkar, Dr.Ragav Venkatesan, Dr.Yilin Wang, Dr. Qiongjie Tian, Yikang Li, Yuzhen Ding, Jaya Gattupalli, Pak Lun Ding, Jashmi Lagisetty.

Lastly, my heartfelt appreciations go to my family. Their love, care and unconditional support encouraged me to complete my Ph.D study.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

How people socialize with other and how they spend their leisure time have been dramatically changed by the impact of social media in the last decade. Rich content distributed throughout our devices, including images, videos, and daily texts, allows users to keep up-to-date on world affairs, current issues, and everyday news. While massive user-generated social media provides interesting resources for many machine learning and computer vision tasks as shown in Fig.1.1, many practical applications still face the challenge of how to efficiently and effectively utilize visual processing, since the user-generated data may pose many challenges for typical algorithms, due uncontrolled nature of the data. To support better visual processing in social media, my research focuses on three main problems. The first focus is to handle variation of scales in computer vision tasks such as face retrieval and eye detection. The second focus is to train reliable classifiers based on partial labels and noisy labels. The third focus is to improve modeling multi-modal data by combining multi-view model and feature selection.

There has been a lot of past research in visual processing with respect to these issues. The first problem involves computer vision applications like eye detection. A method based on independent components analysis (ICA) was proposed in Hassaballah *et al.* (2010). One main limitation of most approaches like this is the requirement of a fixed scale of test images, for example, the assumption that all the images have been preset to a fixed scale. I propose a scale adaptive Eigen approach to address the scale variation on eye detection.

For the second problem, in many applications where training data arises from real-world sources, there may be labeling errors or partial labels in the data. Explicitly correcting label errors for large datasets is expensive and thus may not be done with guarantee. In Xiao

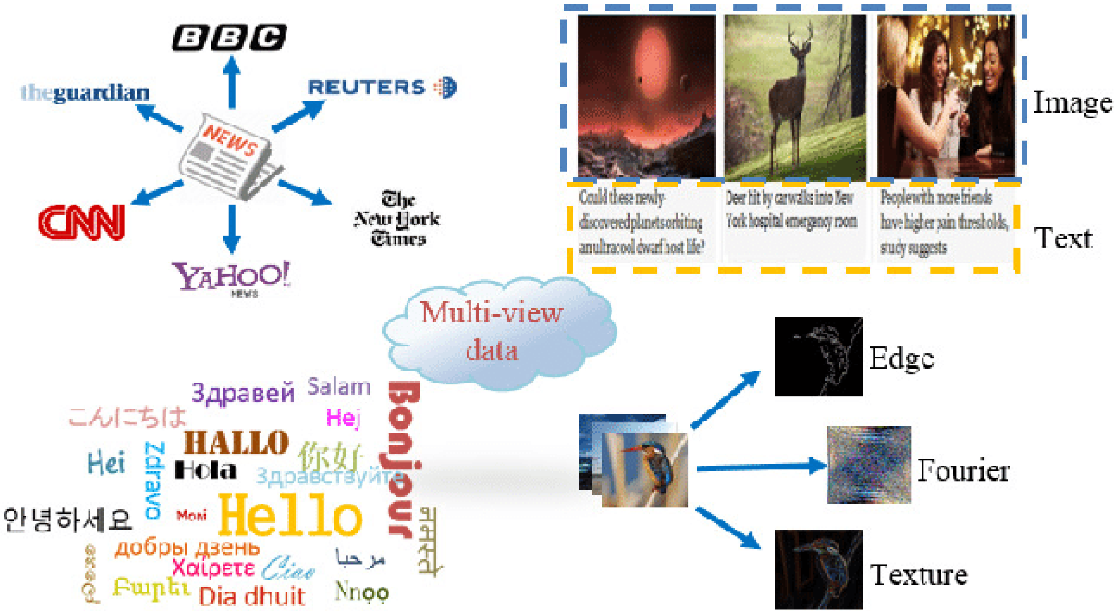


Figure 1.1: Online Data Provide Rich Source for Research.

et al. (2015), a framework was introduced to train Convolutional Neural Networks (CNNs) with only a limited number of clean labels and millions of noisy labels. The relationships between images, class labels, and label noises are modeled with a probabilistic graphical model and integrated into an end-to-end deep learning system. I attempt to address the problem of multi-class learning with noisy labels using a simplified structure: random forest. Other types of methods have also been proposed for applications that lack full labeling information. For example, pairwise matching labels (indicating whether each pair of training sample is the same or different) were used in Guo *et al.* (2013) for learning a discriminative dictionary. I propose a dictionary learning framework that uses less labeling information. This training data will only contain limited sample pairs that should be labeled as the same. For the third problem, most existing work simply combines information from different modal/view when modeling data. I propose a novel, two-view topic model, which interacts with feature selection procedures in learning.

1.1 Eye Detection via Scale Adaptive Eigen Eye

Detecting eyes in images is fundamental for many computer vision applications including face detection, face recognition, and human-computer interaction. Most existing methods are designed and tested on datasets acquired under controlled lab settings (e.g., fixed scale, known poses, clean background, etc.), leaving their performance to be further examined on real-world, uncontrolled images, such as online images. I present an effort on developing a fast and accurate eye detector for online images for which the acquisition condition is unknown and varies from one image to another, resulting in unpredictable background and variable scales for the eyes/faces. The key idea is to develop a scale adaptive Eigen Eye approach, which employs an approximate scale estimated from face detection to modulate the pre-trained Eigen Eye basis in searching for the best match in a test image. The effort also includes building a 2845-image dataset with accurately-annotated eye locations and size. Performing an evaluation using this dataset, in comparison with a few leading state-of-the-art approaches, demonstrates the advantages of the proposed method. Details are shown in Chapter 2.2. Since Singular Value Decomposition (SVD) is essential to many machine learning methods, I further extend the Eigen Eye approach into the more general linear transformed SVD (LT-SVD) problem. The definition of LT-SVD is to solve a new SVD as a linear transform of an already known SVD. In fact, LT-SVD may arise due to the sampling of a dataset and the (approximated) linear transformations between training and testing domains. I present a solution to LT-SVD which, in contrast to traditional methods, achieves both efficiency and accuracy. I also extend the LT-SVD problem to its tensor case and derived solution in the same spirit. My approaches are validated by both theoretical analysis and numerical experiments. Details are shown in Chapter 2.3.

1.2 New Methods for Partial Label and Noisy Label Learning

Discriminative dictionary learning has been widely used in many applications, including face recognition and image classification, where the labels of the training data are utilized to improve the discriminative power of the dictionary. I deal with a new problem of learning a dictionary for associating pairs of images in applications such as face image retrieval. Compared with a typical supervised learning task, in this case the labeling information is very limited (e.g., only some training pairs are known to be associated). Furthermore, associated pairs may be considered similar only after excluding certain regions (e.g., sunglasses in a face image). I formulate a dictionary-learning problem under these considerations and design an algorithm to solve the problem. I also provide a proof for the convergence of the algorithm. Experiments and results suggest that the proposed method is advantageous over common baselines. Details are shown in Chapter 3.1.

Random forest is a well-known and widely-used machine learning model. In many applications where the training data arise from real-world sources, there may be labeling errors in the data. In spite of its superior performance, the basic model of random forest does not consider potential label noise in learning, thus its performance can suffer significantly in the presence of label noise. In order to solve this problem, I present a new variation of random forest - a novel learning approach that leads to an improved noise robust random forest (NRRF) model. I incorporate the noise information by introducing a global multi-class noise tolerant loss function into the training of the classic random forest model. This new loss function was found to significantly boost the performance of random forest. I evaluated the proposed NRRF by extensive experiments of classification tasks on standard machine learning/computer vision datasets like MNIST, Letter, and Cifar10. The proposed NRRF produced very promising results under a wide range of noise settings. Details are shown in Chapter 3.2.

1.3 New Feature Selection Embeded Topic Modeling Method

In the era of big data and the expansion of the internet and social media, analyzing multi-modal data is paramount in gathering information from rich content. Image tweets and micro blog posts, with embedded videos and images, make content more vivid. According to (Chen *et al.*, 2015), multimedia forms attract a larger viewership than text-only posts. On the other hand, many real-world applications involve high-dimensional data in various representations and views that provide related and complementary information. For example, one image can be described by its different views, like color histogram, texture information, SIFT descriptor, and more. As the views are generally high-dimensional and may provide complementary information, extracting the most relevant features from said data is often a necessary step for further analytical tasks. Based on the aforementioned discussion, I aim to embed feature selection into the LDA topic model. While multi-view LDA models can capture correlations between each page view and text/annotations, feature selection pushes the model a step further to make use of the correlation between page views to build a more precise model of multi-modal data. The approach integrates unsupervised clustering and multi-view feature selection into a unifying formulation so that relatedness of the views, clustering of the data, and importance of the features are all simultaneously considered. I focus on how to incorporate the feature selection procedure into the existing multi-view LDA algorithm framework. Details are shown in Chapter 4.

Chapter 2

SCALE-INSENSITIVE DETECTION AND MATCHING IN FACE IMAGE PROCESSING

2.1 Preliminary Exploration Towards Understanding Human Performance of Retrieving Unfamiliar Faces

2.1.1 Introduction

Face retrieval is defined as given a query face image, to find the images containing the same person as in the query image from a large image dataset. The task is in general different from, although related to, face recognition in that it is not necessary to assume a finite set of identities (i.e., subjects) for a given dataset, and the retrieval is done only with respect to the given query image. In recent years, many face retrieval algorithms have been proposed. Wu and Ke proposed a face image retrieval system in Wu *et al.* (2011) using a scalable face representation, where a multi-reference distance approach related to Pseudo-Relevance Feedback (PRF) Manning *et al.* (2008) was used to rank the candidate images using the Hamming signature. In Park and Jain (2010), a method using soft biometrics was described, which employs demographic information and facial marks for improving face image matching and retrieval. In Hu *et al.* (2012), SIFT features were used for initial retrieval, followed by a relevance feedback strategy. There are also efforts dealing with feature selection for face retrieval Dai *et al.* (2011). In general, existing automatic algorithms are either still under-performing or yet to be evaluated on more challenging datasets with images from uncontrolled imaging conditions, indicating much more room for improvement on developing face retrieval approaches.

In the meantime, there have been efforts reported in the literature, attempting to understand human performance on face retrieval or recognition, as it appears human does a better job on such tasks. For example, in Hancock *et al.* (2000) and Adler and Schuckers (2007), human performance on face recognition was compared to automatic algorithms, both drawing the conclusion that human is more accurate than existing automatic algorithms. The study of Furl *et al.* (2002) showed that faces of the same race are easier to recognize by human; and sex effect is studied in Lewin and Herlitz (2002), with the conclusion that women are better at recognizing female faces. Nevertheless, little was revealed on how and why people obtained better performance, and thus it is difficult to derive principles for improving automated approaches.

I set out to explore human performance on the task of retrieving unfamiliar faces. I consider only the task of retrieval of unfamiliar faces (as opposed to face recognition), since such a task would require a subject to derive all the necessary information from a given query image. This would presumably help us to avoid the complicating factors such as memorized identity in the task of recognizing a known person. Hence the study would potentially reveal the types of information that may be employed by an automated system, which would in general perform the retrieval task with information derived only from the query image. To the best of our knowledge, this is a novel problem that has received little attention in the literature. In my study design, to make the retrieval task even more challenging, I employ wild Web face images (as opposed to standard datasets that were used in most existing work). I also design two types of experiments, intended to assess the conscious and unconscious thinking process of the subjects respectively in performing the retrieval task.

2.1.2 Method

Recognizing that a subject may or may not be able to explicitly describe how he/she actually performed the task for a given query image, my basic strategy is to rely on two types of experiments: one mainly based on questionnaire to the subjects to collect their self-reported conscious thinking process during making the query (Experiment 1), and one mainly based on an eye-tracking system for capturing their unconscious search of the visual field of view during making the query (Experiment 2). I elaborate below the experimental protocols and the images used in the experiments.

The face images in my experiments were selected from the Face In the Wild Database Huang *et al.* (2007), a database of face photographs designed for studying the problem of unconstrained face recognition, which contains more than 13,000 images of faces collected from the web. These faces have been automatically labeled using the system described in Berg *et al.* (2005). Since the dataset has 1680 people who have two or more images, I were able to select the images that suit my needs in this study. In particular, I avoid images of well-known celebrities since my goal is to study human performance on retrieving unfamiliar faces. Twenty query images of 20 different people, 10 female and 10 male respectively, were selected for the first experiment; and 40 query images (20 female and 20 male) were used for the second experiment. All selected images were later examined to see if it is unfamiliar before it is used in the analysis.

Additional care was taken in selecting the candidate images for any query image, so as to make the task more challenging. First, people appearing in these images belong to the same gender of the query. Second, I made sure that the candidate images do not have similar clothing style or background to the query image. On the other hand, to ensure the task is better-defined and to support better eye-tacking-based analysis, I also ensured that most of the images have only one dominant face in it.

A software utility was built for the experiments. Each subject who participated in the experiments was asked to log into the study via an interface. After that, the subject can click a menu button to start the experiment, then a query image is shown along with an array of N x N candidate images, as illustrated in Figure 2.1. N is either 4 or 5 in my experiment. The subject then needs to find out which face image (i.e., the target image) in the array belongs to the same person in the query image (making the selection is a simply click of mouse, which takes little time). After that, the subject can click a button to move on to the next query image.

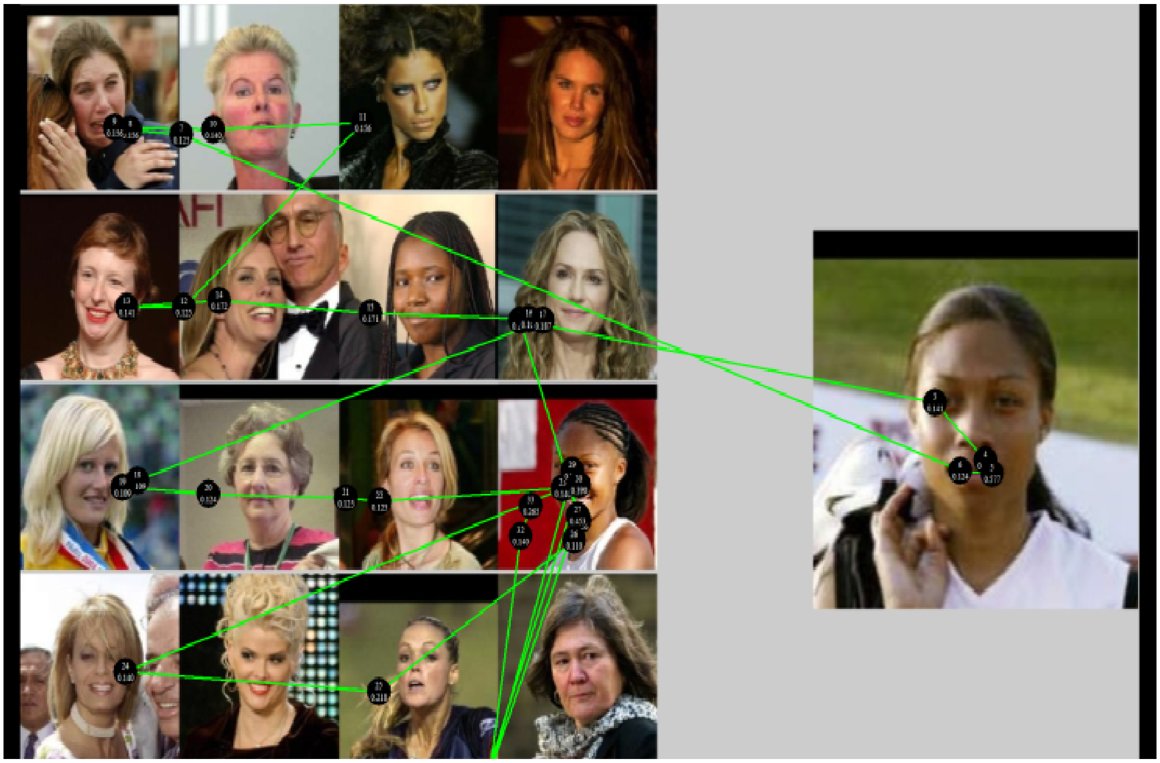


Figure 2.1: Illustrating the Interface

In Experiment 1, after recording the subjects query results (his/her response to each query image, including the time spent on each query image), I asked each subject a set of pre-defined questions and recorded their answers. The questionnaire contains two parts. Part 1 is to confirm the unfamiliarity of the query image. If a subject feels any of the query images have familiar faces, then all the results related to these query images are not considered when I evaluate his/her performance metrics. In Part 2, the following three questions are presented to each subject after the retrieval task is completed:

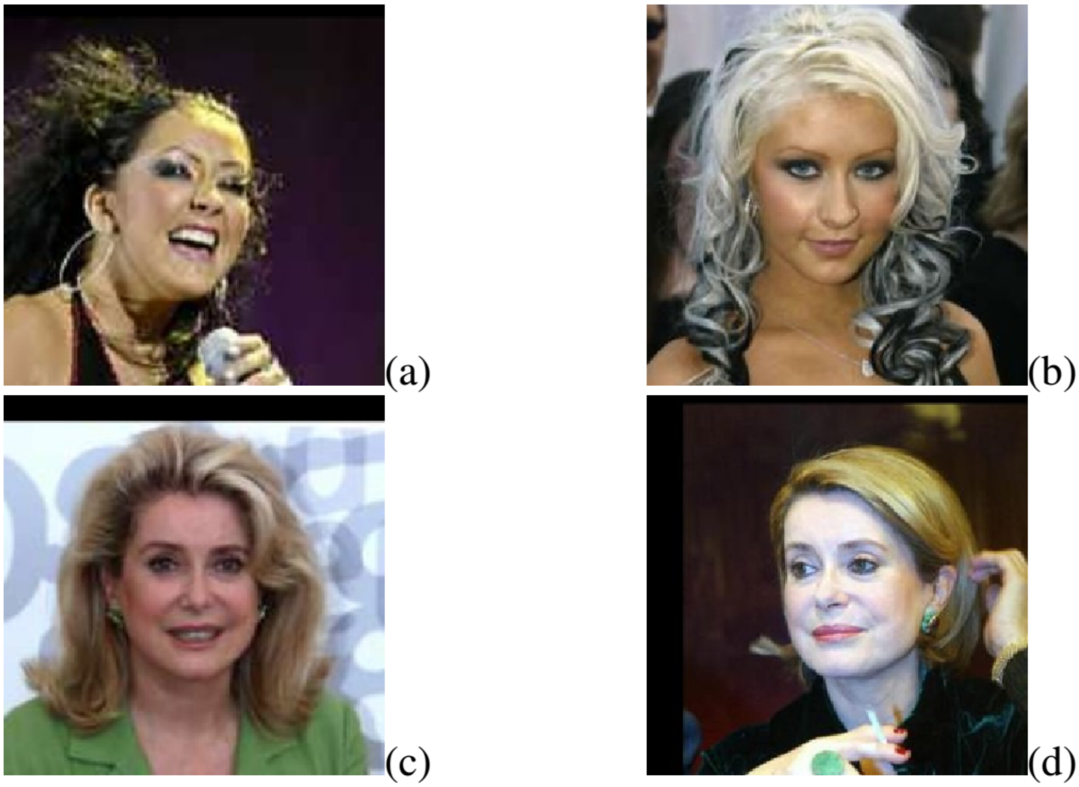


Figure 2.2: Drastic Appearance Changes in Female Face Images.

- Question 1: Do you think color images give more information than grey scale images when you do the retrieval task?
- Question 2: What kind of features do you use or rely on when you retrieve a face?
- Question 3: Rank the importance of the following features in your retrieval: a. Hair style; b. Skin tone; c. Facial features; d. Clothing style; e. Race; f. Age

In Experiment 2, an eye-tracking system was deployed, which records the eye fixation points of a subject doing the retrieval task. The eye-tracking data were time-stamped and thus were later synchronized with the sequence of queries made and thus I can analyze a subjects unconscious search of the visual field for any given query image.

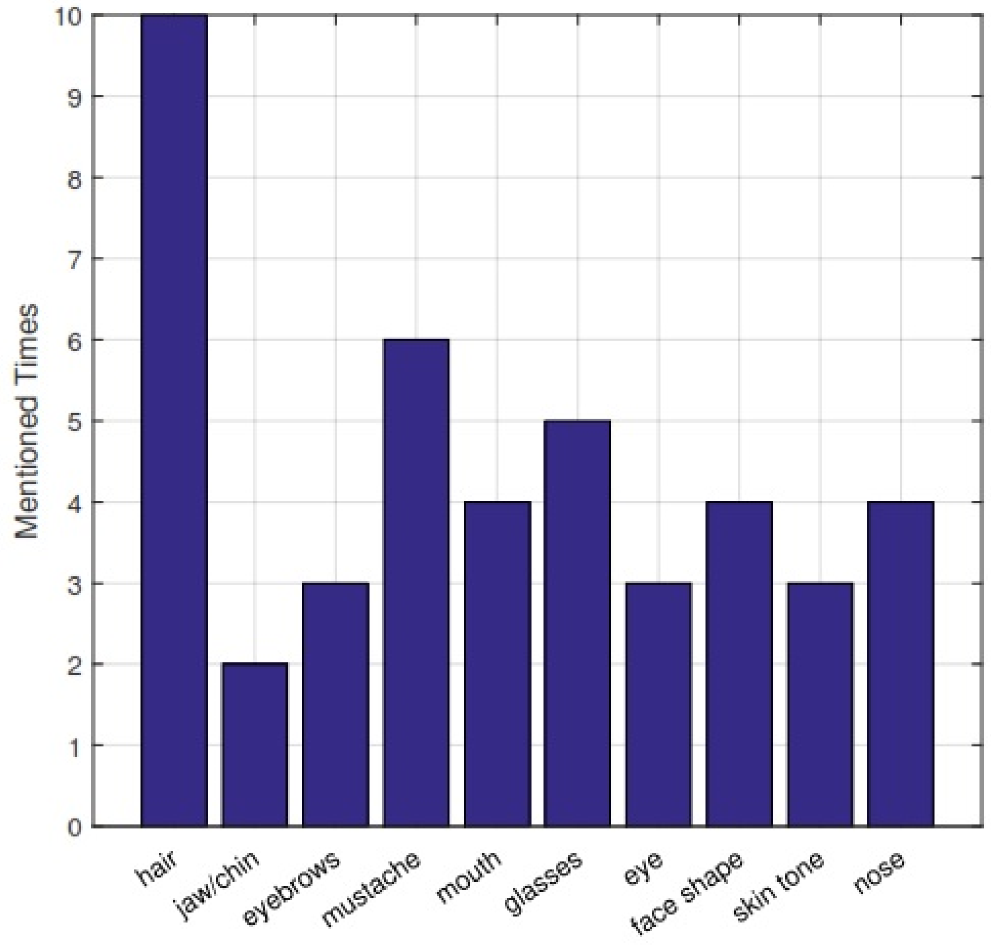


Figure 2.3: Relative Frequency of Features Mentioned by the Subjects

2.1.3 Experimental Procedure

I recruited 11 and 9 subjects for Experiment 1 and Experiment 2 respectively. The age range is from 24 to 31, 2 of them are female and the remaining are male. None of them had prior knowledge about what the query experiments were about and they were trained to perform the task right before the experiments. Each subject was asked to finish 80 sessions of face retrieval in the first experiment. Each query image was shown 4 times in random order, each time with a different candidate set. The 4 candidate sets are:

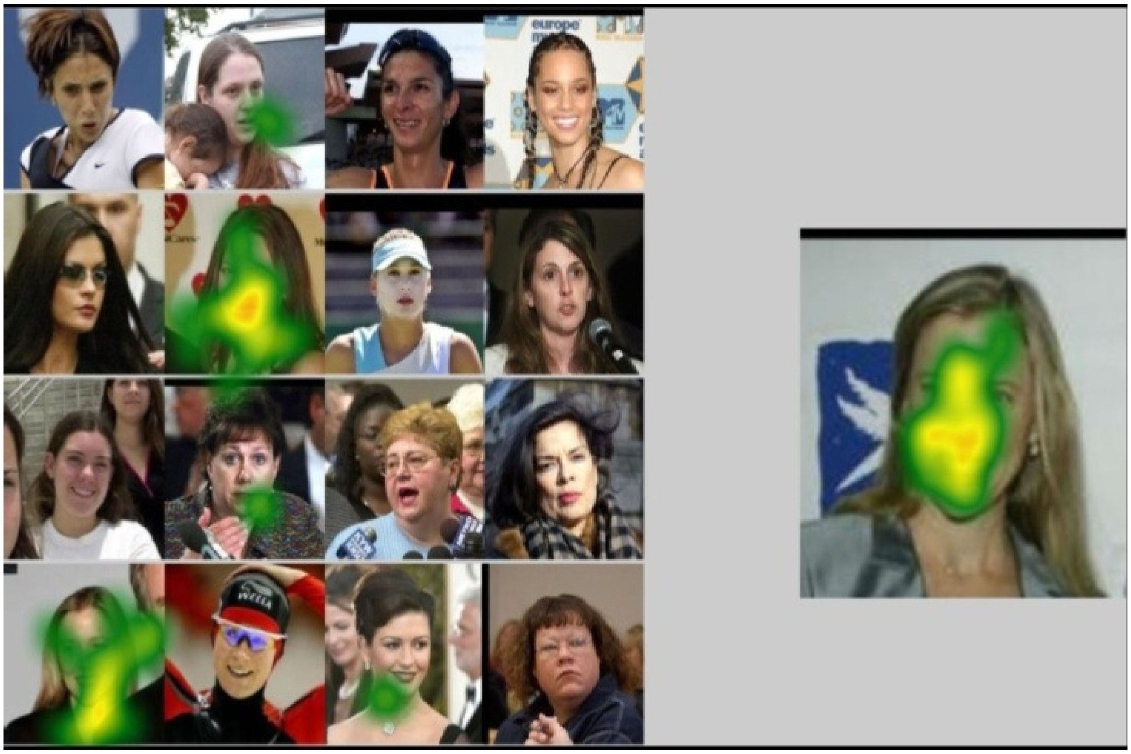


Figure 2.4: Sample Heat Map of One Query Image.

1. Color images with one and only one frontal face image (target image) belonging to the same person as the query image
2. Color images with one and only one off-frontal face image belongs to the same person as the query image
3. Grey scale version of set 1
4. Grey scale version of set 2

The sequence of these 80 sessions is completely random and within each session, the position of the 25 candidate images is randomized too. Additional care was taken to avoid positioning the target image on some special spots (e.g., the top row, where it may be too obvious, or the right-most row, where it is too close to the equerry image on the interface).

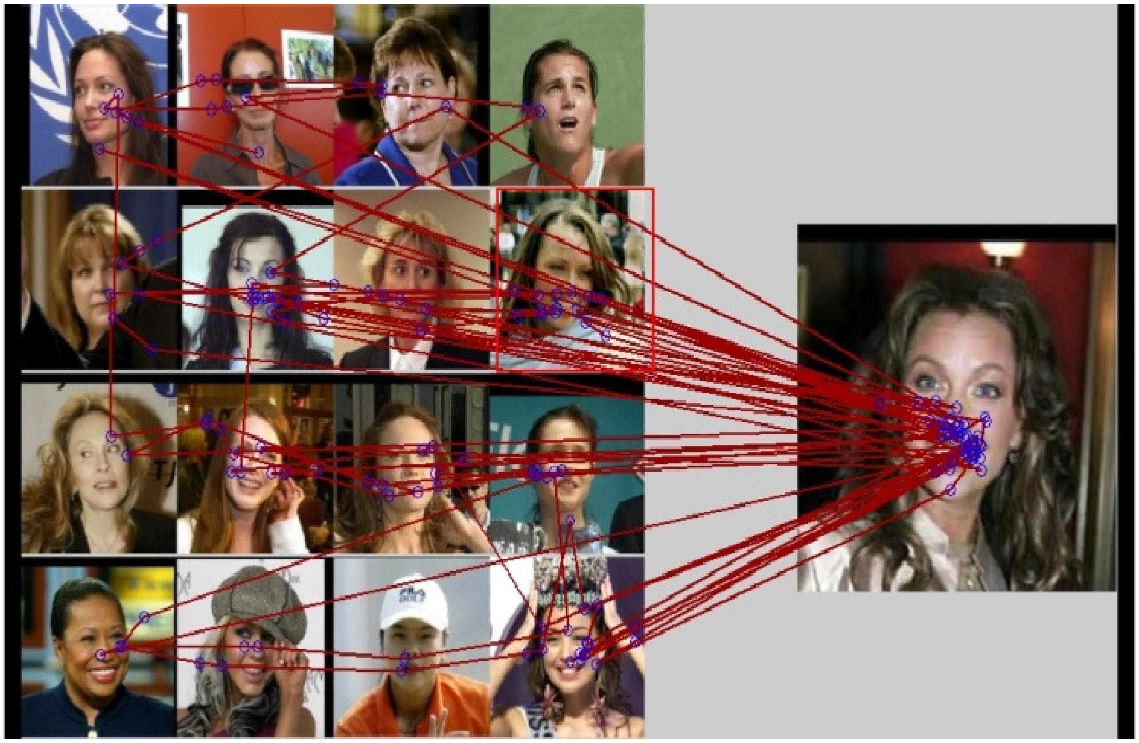


Figure 2.5: A difficult Case with the Busy Gaze Trajectories

In Experiment 2, considering the resolution of the eye tracker, instead of playing 25 images at a time, I reduced the number to 16. After the observations from the first experiment that color images are more supportive to the retrieval task, in the second experiment I used only color images. Hence a total of 40 sessions of face retrieval tasks were done by each of the 9 subjects. I now report the experimental results, which are presented in three different groups: objective performance metrics, subjective/qualitative responses from the participants, and eye-tracking-based experiment and analysis.

2.1.4 Results and Analysis

Objective Performance Metrics in Experiment 1

For each subject, I computed the hit rate and the average time spent on dealing with different types of images, as elaborated below. (For computing the average time spent on certain type of images, I only use results when the query is correctly done.) Overall hit rate: The overall hit rate indicates that the subjects' performances are not ideal. The hit rates fall in to a range from 57.5% 97.5%, with the average of 78.64%.

- Frontal vs. off-frontal faces: The majority (8 out of 11) spent longer time when dealing with off-frontal faces. The average time on frontal faces is 15.2s with STD of 5.2s while it is 17.4s on off-frontal faces with STD of 5.3s. This indicates that pose is an influential factor in face retrieval. An automated algorithm may need to explicitly handle the pose for improved performance.

- Color vs. greyscale faces: The average time spent on color image is 16.0s with STD of 5.6s, while on greyscale image it is 17.4s with STD of 5.5s. 6 out of 11 subjects spent shorter time on color images. This indicates that color images are easier to retrieve than greyscale images. I note that most existing face recognition system actually use only greyscale images.

- Male vs. female faces: The average time spent on male faces is 15.1s with STD of 5.0s, and 19.1s on female faces with STD of 6.6s. 26% more time is spent when dealing with female faces and all the subjects except one spent longer time on female faces than male faces. This indicates that female faces are harder to retrieve. One possible reason is the appearance of a female may change dramatically (e.g., due to new hair style or make-up), as illustrated in Figure 2.2. This suggests that, features that are invariant to common appearance changes in female images should be considered for improving an automated approach.

- Hit rate vs. time spent: As shown above, the hit rate is only 78.64%. Intuitively I may think that the more time people spend on looking at the images, the higher chance that they can select the correct target image. However, interestingly, plotting the average time spent vs the hit rate reveals a wide scatter without any obvious positive correlation among these two metrics.

Subjective Responses

The subjective responses to the questionnaire reveal additional information that either confirms or deepens the understanding of what observed in Sect. 2.1.4. For Question 1: Do you think color images give more information than grey scale images when you do retrieval?, 10 out of 11 subjects responded yes, only one responded no with the explanation that he only looked at facial features. This correlates well with the results of Sect. 2.1.4.

In asking Question 2: What features did you use or rely on when retrieving a face? I intended to learn specific features that human uses on retrieval. Figure 2.3 depicts all features mentioned by subjects and the corresponding number of times they are mentioned. Synonyms are merged. Frequently-mentioned features are: hair, mustache, eye glasses, face shape, nose and mouth. This open question gives us some important hints. For example, most existing automatic face retrieval algorithms do not use hair or mustache as one of the features, while more attention is given to facial components like eyes, nose, and mouths.

For Question 3: Rank the importance of the following features in your retrieval: a. hair style, b. skin tone, c. facial features, d. clothing style, e. race, f. age, ranking scores are from 1-6 while 1 indicates most important and 5 indicates least important. Table 2.1 shows the average ranking scores of these features, lower scores indicate higher priority. Based on these ranking scores, the ranking of the 6 features are: facial features, skin tone, race,

Feature	a	b	c	d	e	f
Score	3.36	2.91	2.73	5.55	3.00	3.45

Table 2.1: Average Ranking Score.

hair style, age, and clothing style.

Experiment with Eye-tracking

Experiment 1 provided some holistic understanding of human performance in retrieving unfamiliar faces, mostly in terms of accuracy (hit rate), time spent, and self-reported important factors. Recognizing the fact that a subject may employ some unconscious process that is not realized by the subject him/herself, I employed eye-tracking in Experiment 2 in order to learn the search behavior of the subjects in retrieving the images. Such experiment may also provide additional data for verifying the self-reported behaviors. In this experiment, an eye tracker tracks the eye fixation and gaze points during the retrieving actions of a subject. The recorded data include: eye gaze points, eye fixation points (duration $\geq 0.1s$), duration of each gaze or fixation points, and special events (when the subject advances to the next query).

The raw data from the eye tracker is the gaze point positions. The gaze points are sampled at a frequency of 100HZ which is dense enough to record accurate eye movement. This high sampling rate also caused crowded gaze points at some positions and thus care needs to be taken to make use of the noisy data. So based on the raw data, fixation points are generated in the following way: in a small radius if the difference of the time stamp of the first gaze point and the last point is more than 0.1s, then the mean position of these points is recorded as a fixation point.

I use both gaze points and fixation points for objective analysis. The candidates that received the least numbers of gaze points are those that were quickly discounted by the subjects. By analyzing such images, I came to the observation that hair color, skin tone and eye glasses contributed to such quick elimination. More specifically, for the 40 sessions, choosing the 5 images with the least gaze points, a total of 5*40=200 images were identified (100 from female candidates and 100 from male candidates). Among these, hair color is the most distinguishing: 64% (female) and 61% (male) rejected candidates having a different hair color than the query image. For female sessions, skin tone is also distinctive: 63% of the candidates have a different skin tone than the query. I also observed 22% of the female candidates are persons wearing eye glasses while the query image does not (or vice versa). For male sessions, the percentages are 41% for the skin tone difference and 45% for with/without eye glasses, respectively.

I can plot the heat map, which shows where the subject looked at during performing the task. Figure 2.4 is a sample heat map for one session, in which I observe that the searching is focused on only a few candidates. The subjects also performed intensive comparisons between query image and candidates. For all the 40 query images, I got 719 fixation points in total, among which 107 points are located around the eyes, 226 around the nose and 386 around the mouth. Surprisingly, I see that eyes are less important than nose and mouth in this searching stage.

Considering both the hit rate and the average time spent on a particular session, I may further define easy cases and difficult cases by thresholding these metrics. With this, I also looked into the ”comparison behavior” of a subject. This behavior is defined as sweeping between the query and candidate images. I found that there is a correlation between the number of sweeps in one session and the difficulty level of the session. The average number of sweeps performed in one session is 56.72. For easy/difficult cases, the average numbers of sweeps are 33.6 and 98 respectively. It is obvious that in the difficult cases, the subjects needed to do more localized checking more often. Figure 2.5 shows a sample from the difficult cases.

Finally, I noticed that not all the candidates in one session received equal comparisons, and the variation can be dramatic. This suggests that most candidates are eliminated quickly by some semi-global features; then intensive comparison is done on a few candidates using more localized feature, and in this stage the nose and mouth are dominant factors that influence the final decision. Such a finding may lead to a staged design of new automated retrieval approaches: semi-global-feature-based elimination followed by localized-feature-directed refined comparison.

2.1.5 Conclusion

I carried out experiments to understand human performance in retrieving unfamiliar faces. In addition to measuring the hit rate while documenting other performance metrics, I employed eye-tracking to capture the search patterns of the subjects. The results, while confirming some intuitions, also revealed new clues as to what are important in humans retrieving action, providing insights into new ways of improving automated approaches.

2.2 Scale-adaptive Eigeneye for Fast Eye Detection in Wild Web Images

2.2.1 Introduction

Eyes are among the most salient facial features in facial images. Accordingly, eye detection is often a fundamental module in many applications involving facial image analysis. For example, many face recognition engines rely on eye detection (explicitly or implicitly) for pre-aligning the face images before any training/testing algorithm is applied. Some applications (like creating tactile facial images Wang *et al.* (2008)) require high-precision eye detection. Although it has been intensively studied over the past years, fast and accurate eye detection remains a challenging task especially if uncontrolled imaging conditions are considered. Lacking an accurately-annotated dataset of uncontrolled real-world images has prevented a direct comparison of various approaches proposed thus far, and hence hindering my understanding of the real performance of these algorithms. My study in this chapter attempts to bridge these gaps.

I start with reviewing some most recent approaches in the literature. Such recent studies typically report significant improvement over earlier approaches (Corrochano (2005),Press *et al.* (1992),Wang *et al.* (2005a),Tang *et al.* (2005),Wang *et al.* (2016)), and thus my review will not go back to those earlier methods. A method based on independent components analysis (ICA) was proposed in Hassaballah *et al.* (2010). This approach was reported to achieve a detection rate of 97.3% on 1500 images from the FERET dataset. One main limitation of this approach is the requirement of a fixed scale of test images, for example the assumption that all the images have been pre-aligned to a fixed size (as is the case in FERET). Another state-of-the-art approach is the feature-versus-context approach in Ding and Martinez (2010). This approach was reported to achieve an average detection rate of 97% on the AR, XM2VT and ASL datasets. As will be demonstrated later, one key drawback of this method is its inaccuracy in detecting the correct size of the eyes. In addition, there are also some other recent efforts on eye detection. For example, in Park *et al.* (2010), textural characteristics of eye regions and non-negative matrix factorization (NMF) based image reconstruction are considered for eye detection. In Gan and Liu (2010) and Ren and Jiang (2009), eye detection based on rank order filter is introduced. All these approaches reported similar good performance on either standard or proprietary datasets.

Despite the promising result of the above methods, they were all based on standard face datasets with well-aligned and normalized face images or proprietary data. In this study, by manual annotation, I built a dataset of 2845 images taken from the Face Detection Dataset and Benchmark (FDDB) Jain and Learned-Miller (2010) from the Faces in the Wild project (vis-www.cs.umass.edu/lfw/). I have manually labeled the eye region and eye center location as the ground truth. Inspired by the success of eigen analysis in face recognition (e.g. Quintiliano and Santa-Rosa (2003), Poon *et al.* (2009), Park *et al.* (2010)), I adopt an EigenEye approach. Furthermore, recognizing the drastic scale variation in the wild images, I propose a scale-adaptive scheme, in which EigenEye trained from high-resolution eye images are updated according to an estimated eye scale (from face detection) for eye detection. These considerations result in a fast (due to mostly linear eigen space projections and distance computation) and robust (achieved by eigen analysis and scale adaptivity) eye detector, which outperforms other state-of-the-art techniques on this challenging set of wild Web images. The dataset will be made public to support further fair comparative studies by researchers working on this topic.

2.2.2 Proposed Approach

In this section, I introduce a scale-adaptive EigenEye scheme for eye detection. The objective is to achieve robustness of detection in wild Web images through eigen analysis and scale adaptivity. The main idea for achieving scale adaptivity is to resize the eigen bases obtained in the training stage, based on an estimated eye scale. I first outline the general framework of the method in Section 2.2.2, and then elaborate the scale-adaptive scheme in Section 2.2.2, with the overall eye detection algorithm summarized in Section 2.2.2.

The General Framework

The proposed EigenEye approach consists of two stages: training and detecting. In the training stage, a training set of high-resolution eye images are collected. (In the current study, 861 such images of size 120×160 are used, although these can be updated as needed.) These images are used in eigen analysis to extract the top eigen bases, i.e. EigenEyes. In the current study, the first 60 EigenEyes are kept. Every training eye image is then projected into this subspace spanned by EigenEyes, i.e. EigenEye space. In the detecting stage, I first perform face detection to obtain the candidate face region. Then I search for eyes throughout the face region as the best match to the training eyes in the EigenEye space. Obviously, for wild Web images, the size of the eye in the test image can be very different from that of the training set. Hence the matching needs to be done only after both the test image and the EigenEye space have been normalized in the same scale. To this end, I propose to modulate the eigen space by adapting the EigenEyes based on the size of the detected face region. Adapting the EigenEyes to a test image instead of normalizing the test image with respect to an EigenEyes is motivated by the fact that the eyes to be detected will be in general at lower resolutions than training eye images or EigenEyes. Therefore, adapting the high-resolution EigenEyes to a lower-resolution test image can better ensure the matching is done with images at the same imaging resolution. I will elaborate the approach in the Section 2.2.2.

Scale-Adaptive EigenEye for Eye Detection

the s be the size ratio between the current input image z anScale-Adaptive Eigeneye (SAE) Eigen analysis or principal component analysis (PCA) is a mathematical procedure that projects a set of observations of possibly correlated variables into a set of linearly uncorrelated variables called principal components. To be specific, let $X = [\hat{x}_1, \hat{x}_2, ...\hat{x}_N]$ be the data matrix where $\hat{x}_i = x_i - \mu$ is the centralized vector of the i-th sample $x_i \in R^D$so that $\hat{x}_i = 0$. PCA takes the eigenvectors $V = [v_1, v_2, ...v_K]$ related to the K biggest eigenvalues $_i, 1 \leq i \leq K$ of the covariance matrix $C = XX^T$ as its orthonormal transform matrix. Then, a D dimensional sample x is converted into a K dimensional subspace by $y = V^T(x - \mu)$ which can be explained from two aspects.

PCA-based face recognition and detection has been intensively studied Poon *et al.* (2009),Liao and Lin (2005), where each PCA eigenvector v_i is called eigenface. Eigenface recognition is in general very fast, since projection into eigenfaces is just a linear transformation. Similar idea can be used to develop an EigenEye scheme for eye detection. However, as discussed above, the limit of an EigenEye detector is that the resolution of test image should be the same as that of the EigenEyes. A traditional way of scale adaption is to resize the test image before EigenEye projection.

It is intuitive to think that instead of calculating PCA only once at the original scale and enlarge every test image, I can compute PCA eigenvectors every time at the scale of a test image. To do this, I first resize the training data to the scale of the test image, then perform PCA. One obvious drawback of this method is that the computational cost will be very high if I use high resolution and large number of training data (as is often the case) which make it impossible for practical applications.

To alleviate the above problems, I propose to adapt the EigenEyes to the (in general) smaller-scaled test image. The Nystrom method (Burges (2005),Fowlkes *et al.* (2004)) gives a way of calculating the approximate eigenvectors and eigenvalues of a matrix K, using those of a submatrix A. Inspired by this, I give a solution in a more general situation: approximating eigenvectors and eigenvalues of a matrix A with scale s, using those of a matrix K with scale t, $t > s$. Here I introduce an linear re-scale operator$D$, given training data matrix X, the eigen decomposition of XX^T is $V\Lambda V^T$. Then $X_s = DX$ is the re-scaled training set, from the eigen decomposition of XX^T, I can get:

$$(DX)(DX)^T = DXX^TD^T = DV\Lambda V^TD^T \tag{2.1}$$

If the columns of DV are orthogonal, then I get exactly the new eigenvectors I want, but in general cases, they are not. At this point, let

$$L = DV\Lambda^{1/2} \tag{2.2}$$

then the SVD (singular value decomposition) of L would be $L = V_L\Lambda_L {S_L}^T$, now I rewrite

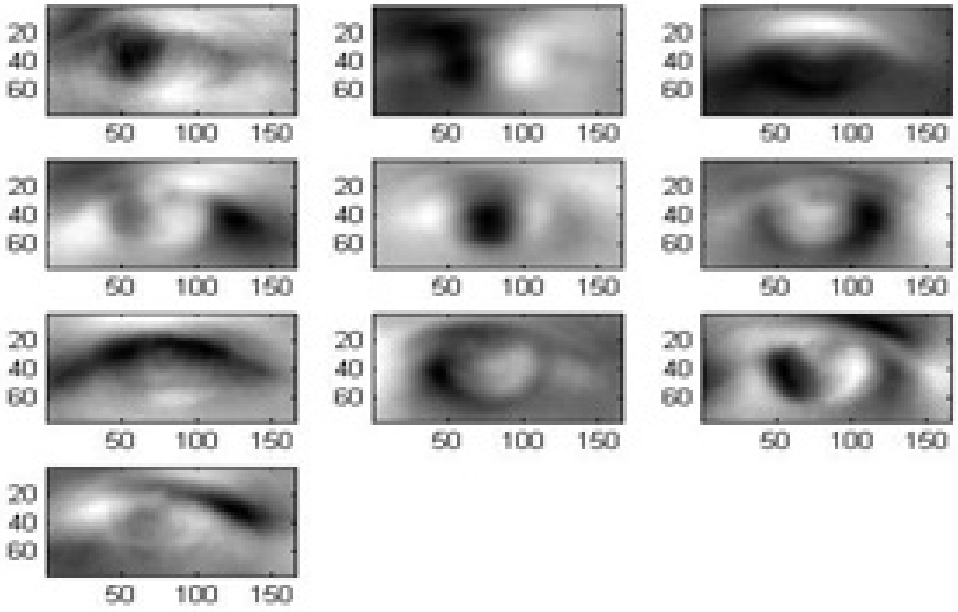


Figure 2.6: Examples of Eigeneye Basis.

2.1:

$$DV\Lambda V^T D^T = L\Lambda L^T = V_L \Lambda_L^2 V_L^T \tag{2.3}$$

Since the columns of V_L are orthogonal to each other, I already got the new eigenvectors. The above discussion gives the exact eigen decomposition of re-scaled covariance matrix. Based on this, I can derive a straightforward way to get the approximate eigeneye basis. To be specific, I only need to perform PCA on the original training set X once, and select first K eigenvectors (eigeneye basis)and corresponding first K eigenvalues. For each eigeneye basis $v_i, 1 \leq i \leq K$of V, I resample it to scale s by multiply the operator$D$ to obtain a V_s, then I construct L using 2.1 and do SVD to get re-scaled eigeneye basis.

Figure 1. Examples of Eigeneye basis. First 10 eigen basis, correspoding to largest 10 eigenvalues.

Eye Detection: the Complete Procedure

With the previous preparation, I now describe the complete procedure for my eye detector. This involves the following five steps of processing for any given image to be tested.The flow chart is shown in Fig.2.7.

Step 1. Face detection using OpenCV.

Step 2. Scale estimation. Suppose that the size of detected face image is $m \times n$. Its scale compared to my training image is $s = n/N$, where N is width of the training face images. Then the size of candidate eye bounding box is estimated as $x \times s \times y \times s$, where $x \times y$ is the size of eyes in the training set.

Step 3. Eigeneye adaptation. Resample the EigenEyes V by scale s following Section 2.2.2. The dimension of the modified eigeneye basis ,i.e. each column of V_s, is $x \times s \times y \times s$.

Step 4. Computing matching scores. I use sliding window to get the matching sore of all candidate eye blocks to training eye region images. Since the OpenCV face detection result can't always cover precisely the entire face region, I do a scale search instead of using one fixed scales. Repeat step 2 to 4 on different scales. For each window, several matching score will be recorded and each corresponding to a certain scale.

Step 5. Detecting true eye region from the candidate blocks. The matching scores of all locations in the search range are used to identify the final eye locations.

2.2.3 Experiments and Results

I now describe the evaluation of the proposed scale-adaptive EigenEye method (SAE) and compare with several state-of-the-art methods. First, I compare the reconstruction errors of my proposed SAE method with two other general PCA methods. Second, two of the most recent eye detection approaches are chosen for comparison, namely, the ICA based method (ICA) Hassaballah *et al.* (2010), and the feature-versus-context (FVC) method Ding and Martinez (2010). I also included the eye detector provided by OpenCV in the comparison for its wide availability. I built a dataset of 2845 images based on the FDDB face database Jain and Learned-Miller (2010) which contains faces with different scales and background, as illustrated in Fig. 2.8. The eye detection performance is evaluated by two metrics, i.e. precision of eye center and precision of detected bounding box.

Ground Truth

To provide a fair comparison using wild Web images, efforts were devoted to manually annotate each of the 2845 images. For the dominant face in a given image, the eyes are manually marked by 4 points: the center of the right eye center, the center of the left eye, the upper-left corner and lower-right corner of the left eye. Fig. 2.9 illustrates some of the examples (for the cropped face region only, for better visualization).

Accuracy in Detecting the Eye Center

For a given face image, each of the approaches reports the two detected eyes as two bounding boxes whose center is estimated eye center. Then I use Euclidean distance to measure the distance between the ground-truth eye centers and detected eye centers. If this distance is smaller than one third of the width of the ground-truth bounding box (which is roughly half of the pupil size), it is deemed as a hit. The hit rate based on this protocol is given in Table 2.2 for comparison. From Table 2.2, it can be seen that the OpenCV eye detector performs poorly on this dataset. The proposed method (SAE) is comparable to FCV in the precision of eye center. Note that the OpenCV face detector based on Adaboost classifiers

Method	SAE	FVC	ICA	OpenCV
Accuracy	97.36%	97.25%	92.4%	29.3%

Table 2.2: Eye Center Hit Rate.

and Harr-like features achieved 91% detection rate on my wild Web face image dataset. Accordingly, in order to avoid introducing the inaccuracy of the face detector to the eye detectors, in this chapter, the eye detection accuracy is only calculated on those images with successful face detection.

Accuracy in Detecting the Eye Box

An accurate eye detector should also precisely estimate the size of the eye, i.e. obtaining a proper bounding box. One possibility is to compute the following metrics that measure the similarity between the ground-truth and the detected bounding boxes:

$$precision = \frac{(Groundtruth\ area \bigcap Detected\ box\ area)}{Detected\ box\ area}$$

While these metrics are intuitive, they are not very effective for my study. For example, the precision can still be 100 if the detected box is overly enlarged and fully cover the ground-truth box. Hence I also introduce two other complementary measurements, i.e. the ratio area and Jaccard similarity coefficient (JSC):

$$ratio\ area = \frac{Detected\ box\ area}{Groundtruth\ area}$$

$$JSC = \frac{(Groundtruth\ area \bigcap Detected\ box\ area)}{(Groundtruth\ area \bigcup Deteted\ box\ area)}$$

Method	SAE	FVC	ICA
Precision	55.1%	43.55%	33.54%
ratio area	0.69	2.98	2.45
JSC	48.86%	25.56%	22.97%

Table 2.3: Precision and Ratio of Eye Bounding Box.

The average performance numbers of the competing approaches are given in Table 2.3. (The OpenCV eye detector is not reported here because of its extremely low performance.) Table 2.3 shows that my method is consistently superior to other methods on the precision and JSC of bounding box. Besides, the area of its bounding box is also close to the ground truth.

Summary of the Results and Discussion

The results presented in Table 2.2 and Table 2.3 suggest that the proposed SAE approach is slightly better than FVC in terms of locating eye center, while being much better than FVC in terms of estimating eye bounding box.Both SAE and FVC are better than other approaches. Hence overall the proposed method is deemed as the best among all four approaches. Fig. 2.10 provides an example illustrating the results from the approaches evaluated in this study. Another benefit of the proposed method is its speed performance. I ran all the approaches on a PC with AMD A8-3500M APU 1.50GHz and 8.00GB RAM. In terms of the average execution time for one image, my method spends only 45% and 50% of the times needed by the FVC approach and the ICA approach respectively.

2.2.4 Conclusions and Future Work

I presented an eye detector using scale-adaptive EigenEyes for wild Web images. I built a dataset with manually annotated ground truth for evaluating competing algorithms. Based on the dataset, the comparative experiments have demonstrated the advantage of the proposed method, in terms of both overall accuracy and speed performance. There are a few directions for further exploration. Firstly, the current version of the proposed method relies on only a small training set of 861 eyes. Conceivably, its performance may be further improved by using a better training set. Secondly, my method currently does not use any color information. Incorporating color may help further improve the performance especially for reducing false detection. There are other many eigen problems in pattern recognition applications De Bie *et al.* (2005), and one future direction is to explore the proposed idea for achieving scale-adaptivity in such problems.

2.3 On Linear Transformed Singular Vector Decomposition Problem

2.3.1 Introduction

A lot of machine learning methods can be reduced to Singular value decomposition (SVD) problem (Skillicorn, 2007) or the related Eigen value decomposition (EVD), where the Singular or Eigen vectors of training data matrix (or kernel/covariance matrix) are assumed to carry the major pattern information. Then testing samples are projected into the subspace spanned by these Singular or Eigen vectors for further recognition tasks. However, in practice, test samples may lie in different data domains than that of training samples. As a result, the subspace derived from training domain may not fit testing domains, and I may expect some adaptation to make use of the training result for another testing domain. While this belongs to a general topic of domain adaptation (Ben-David *et al.*, 2010; Pan *et al.*, 2011), in this chapter, I focus on the adaptation of SVD/EVD when testing domain can be derived from training domain by linear transforms. This leads to the linear-transformed SVD/EVD (LT-SVD/LT-EVD) problems as clearly defined later. Since the EVD of symmetric matrix is quite related to SVD (Van Loan, 1996), I use the term LT-SVD for the whole spectrum of problems. In fact, the solution to a LT-SVD problem can be directly extended to its peer LT-EVD problem. The result and application of LT-SVD problem is quite general. First, SVD and EVD lies in the heart of many machine learning methods (Ben-David *et al.*, 2010; Hastie *et al.*, 2015; Shamir, 2015; Guo *et al.*, 2016), especially in computer vision domain (Agrawal and Khatri, 2015; De *et al.*, 2015; Mahmud *et al.*, 2015; Zhou *et al.*, 2016; Mehta *et al.*, 2014). Second, a lot of transforms among data domains can be approximated well by linear transforms, e.g. resize, rotation, translation.

Lets explain the above LT-SVD problem by an example of popular PCA-based face recognition (Turk and Pentland, 1991; Gottumukkal and Asari, 2004) where eigenface is derived as the left Singular vectors of the training data matrix (X) (or the Eigen vectors of covariance matrix). Test images are projected into the subspace of eigenface for recognition. However, test images can have various sizes due to different acquisition condition. Since image resize can be well approximated by linear transforms, e.g. bilinear interpolation, the testing domain can be characterized by the data matrix of $L_L X$, where X is the data matrix with training images in columns, and L_L is the linear resize operator. I may want to adopt the SVD of X for the SVD of $L_L X$(Fig.**??**), which is a typical LT-SVD problem.

Three solutions to the LT-SVD problem are widely adopted in practice. Without loss of generality, let X be the training data matrix which is transformed into testing domain by L_L. The first solution is called re-SVD which transforms all training samples into the testing domain by L_L, and re-compute the SVD of $L_L X$. Despite its preciseness, re-SVD introduces heavy computation to get $L_L X$ for a large data set. The situation can be even

Table 2.4: Traditional Solutions to TYPE-I LT-SVD Problem

Method (Abbrev.)	Comp.	Accur.	Description
Re-SVD (RS)	55	✓	SVD($L_L X$)
Inverse-Domain (ID)	✓	55	$\widetilde{L}_L^+ Y$
Base-Transform (BT)	✓	55	$L_L U$

worse in ensemble methods, e.g. rotation forests (Rodriguez *et al.*, 2006), where SVD is required for each sampling subset. The second is called inverse-domain which applies an approximate inverse of L_L, e.g. $\widetilde{L}_{L^+}$, to convert a test sample into the training domain so that the original SVD of X can be used for further application. While it is fast, inverse-domain suffers from the loss of accuracy when L_L is not is invertible, e.g. down-size transform. The third is called base-transform which directly transforms the Singular vectors of X, e.g. U, by L_L, and use $L_L U$ (normalized) to approximate the SVD of test domain. Despite its simplicity, the $L_L U$ is not orthogonal. So a large approximation error might arise in base-transform, which will be clear in later experiments. The three solutions to LT-SVD and their pros and cons are summarized in Table 2.4.

Since all the methods in Table 2.4 have limitations, I might want a new solution with both efficiency and accuracy. I start at the very simple intuition: can I get the linear transformed SVD directly from the original SVD? Luckily, the answer is YES. The main idea is that the data matrix in testing domain equals the linear transform of the data matrixs SVD in training domain. As a result, the LT-SVD can be derived on the original Singular vectors/values. The method is efficient, for it directly works on the Singular vectors; it is accurate, since I can have truncated SVD on arbitrary accuracy.

My work contributes in two aspects. First, I present a solution of LT-SVD problem with both efficiency and accuracy. Second, I also extend the result of LT-SVD into tensor case. My solution can be adopted by many subspace learning methods (Skillicorn, 2007; Chandrasekaran *et al.*, 2011; Rahmani and Atia, 2016; Mardani *et al.*, 2015) and their ensemble extension (Rodriguez *et al.*, 2006) in the case of data transform or sampling. One related work is the online SVD (Brand, 2003; Jin *et al.*, 2016),while both works try to infer the new SVD from a related matrix, the online SVD focus on the enlargement by new rows or columns which cant be represented by linear transforms. Another quite related work is the Nystrom EVD in kernel learning (Burges *et al.*, 2010) which approximates the EVD of a rank-r kernel matrix K by that of a $r \times r$ full rank sub-matrix K_{rr}. Since K can be derived by a linear transform of K_{rr} (Example 4), Nystrom EVD is just a special case of the transformed SVD. The structure of the chapter is as follows. The problem and notation definition is in Section 2.3.2 followed by the transformed SVD for matrix and the extension to tensor case. Experiments are given in Section 2.3.3 and I finally concluded in Section 2.3.4.

2.3.2 Preliminaries and Problem Definition

The SVD (Van Loan, 1996) of a matrix is $X = U\Lambda V^T = \Sigma_{i=1}^r \sigma_i u_i v_i^T$, where $r = rank(X)$, $U = [u_1, ..., u_r]$ and $V = [v_1, ..., v_r]$ are orthonormal left/right Singular vectors, and $\Lambda = diag([\sigma_1, ..., \sigma_r])$is a diagonal matrix composed of ordered non-negative Singular values, i.e. $\sigma_i \geq \sigma_{i+1} >= 0$. I use SVD(X) for the triplet of (U, V, Λ). The SVD can be applied to any matrix and provides insight on matrix structure with important applications in data mining (Skillicorn, 2007). Truncated SVD is to approximate X with the first t leading Singular values and vectors, i.e. $X \sim U_t \Lambda_t V_t^T = \Sigma_{i=1}^t \sigma_i u_i v_i^T$. In fact, truncated SVD provide the lowest approximation error among all rank-t matrices, i.e. $\|X - U_t \Lambda_t V_t^T\|_2^2 = \Sigma_{t+1}^r \sigma_i^2$ (Van Loan, 1996). The EVD of a symmetric matrix S always exists as $S = U \Sigma U^T$, where the matrix U is composed of orthogonal Eigen vectors and the diagonal Σ is composed of Eigen values (Van Loan, 1996). I also use EVD(S) for the

2-tuple of (U, Σ). Similarly, truncated EVD is to approximate S with the first t leading Eigen values and vectors, i.e. $S \sim U_t \Sigma_t U_t^T$.The SVD and EVD are quite related. In fact, the left/right Singular vectors U and V of SVD(X) is exactly the Eigen vector of XX^T and $X^T X$, and the Singular value is the square of Eigen values. So I focus on LT-SVD problems, whose solution can be directly extended to its peer LT-EVD problems. In this chapter, I are interested in the EVD of positive semi-definite (PSD) matrices, i.e. $S = XX^T$, which are quite popular in machine learning, e.g. kernel matrix and covariance matrix. In fact, $S = (U\Sigma^{1/2})(U\Sigma^{1/2})^T$ is a special case of this decomposition. The LT-SVD problems are defined below. Since the linear transform of X can have 3 cases, i.e. $L_L X$, XL_R^T, and $L_L X L_R^T$, I have 3 types of transformed SVD problems:

- Type-I LT-SVD: SVD(X)$\rightarrow$SVD($L_L X$). The meaning of the above notation is that SVD($L_L X$) should be directly derived from SVD(X) rather than re-computation of $L_L X$. Similarly, I have the following two types.
- Type-II LT-SVD: SVD(X)$\rightarrow$SVD(XL_R^T).
- Type-III LT-SVD: SVD(X)$\rightarrow$SVD($L_L X L_R^T$)

Similar to LT-SVD, there are 3 types of LT-EVD for PSD matrix which can be written as $S = XX^T$.

- Type-I LT-EVD: EVD(XX^T)$\rightarrow$ EVD($L_L X X^T L_L^T$).
- Type-II LT-EVD: EVD(XX^T)$\rightarrow$EVD($XL_R^T L_R X^T$).
- Type-III LT-EVD:EVD(XX^T)$\rightarrow$EVD($L_L X L_R^T L_R X^T L_L^T$).

Although the three traditional solutions in Table 2.4 are defined for Type-I LT-SVD, they can be easily extended to the other LT-SVD or LT-EVD problems in the same spirit.

Transformed SVD Problem

In this section, I introduce my solution to LT-SVD which enjoys both efficiency and accuracy.

Theorem 1. (Type-I LT-SVD). Let $SVD(X) = U\Lambda V^T$. The SVD of$L_L X$, i.e. $U_L \Lambda_L V_L^T$, can be derived from SVD(X) as follows. Let U_L and Λ_L be the left Singular vector and Singular value of SVD($L_L U \Lambda$) whose right Singular matrix multiply by V will lead to V_L.

Proof: From SVD(X), I get $L_L X = L_L U \Lambda V^T$. Let $U_1 \Lambda_1 V_1^T$ be the SVD of $L_L U \Lambda$. Then I have:

$$\begin{aligned} L_L X = (L_L U \Lambda) V^T = (U_1 \Lambda_1 V_1^T) V^T \\ = U_1 \Lambda_1 (V_1^T V^T) = U_1 \Lambda_1 (V V_1)^T = U_L \Lambda_L V_L^T \end{aligned} \tag{2.4}$$

where $U_L = U_1$, $\Lambda_L = \Lambda_1$, and $V_L = V V_1$. Now I need to prove $U_L \Lambda_L V_L^T$ is the SVD of $L_L X$. Since U_L and Λ_L come from SVD($L_L U \Lambda$), they must satisfy the condition of SVD. It remains to show $V_L = V V_1$ is orthogonal, which is quite clear since both V and V_1 are orthogonal matrix. Usually, I only have rank-t truncated SVD as X $X_t = U_t \Lambda_t V_t^T$. Following theorem 1, I can derive SVD($L_L X_t$) from $U_t \Lambda_t V_t^T$ to approximate SVD($L_L X$). The complexity and accuracy of this approximation are discussed below.

Remark 1. Let the size of X be $M \times N$. Then size of U_t and Λ_t from truncated SVD is $M \times t$ and $t \times t$ respectively. Let the size of L_L be $D \times N$. So $L_L X$ is a $D \times M$ matrix. From the work in (Skillicorn, 2007), the complexity of SVD($L_L X$) is $O(DN \cdot \min(D, N))$. From Theorem 1, to derive SVD($L_L X_t$) from $U_t \Lambda_t V_t^T$, I need to compute SVD($L_L U_t \Lambda_t$) whose complexity is $O(Dt \cdot \min(D, t))$.When $D > N$, the complexity ratio between my method and direct SVD($L_L X$) is $O(t^2 / N^2)$; when $N > D$, the time ratio is $O(t^2 / (N \cdot D))$. With $t \ll \min(D, N)$ the computation is greatly reduced by approximate SVD($L_L X$) with my method.

Remark 2. Since Singular vectors are what I need in many machine learning methods, one may ask: How the approximation of SVD($L_L X$) by SVD($L_L X_t$) affects the Singular vectors? Let (u'_i, v'_i) be a pair of Singular vectors of $L_L X_t$ with $1 \leq i \leq t$ ($rank(X_t) = t$); Let (u_i, v_i) be its corresponding Singular vectors of $L_L X$ and $(\widetilde{U}_i, \widetilde{V}_i)$ are the matrix of the remaining Singular vectors. From Theorem 8.6.5 in (Van Loan, 1996), there exist two vectors p and q so that $u'_i = c(u_i + \widetilde{U}_i p)$ and$v'_i = d(v_i + \widetilde{V}_i q)$, where c and d are normalization constant. Due to the orthogonality among Singular vectors, the inner product of (u'_i, u_i) is $(1 + \|p\|_2^2)^{-1/2}$ and that of (v'_i, v_i) is$(1 + \|q\|_2^2)^{-1/2}$. The theorem shows $\|[p; q]\|_F \leq 4\|E\|_F/\delta_i$ where $E = L_L X - L_L X_t$ and $\delta_i = \min_{j \neq i} |\sigma_i(L_L X) - \sigma_j(L_L X)|$ with $\sigma_i(L_L X)$ be the i-th Singular value. Therefore, the accuracy of Singular vectors is controlled by the approximation error due to SVD truncation (E) and Singular-value gap (δ_i).

Example 1. (Resize PCA) Here is an application of type-I LT-SVD. Let a data matrix be $X = [x_1, x_2, ..., x_N]$, where $x_i \in R^M$ is the i-th sample which is already centralized to zero-mean. Then PCA vector is the left Singular vector of X. Suppose a test sample lays in a lower dimension which can be approximated by a down-size linear operator L_s. Then the PCA in the test domain is SVD($L_s X$), i.e. a type-I LT-SVD problem.

Corollary 1. (Type-II LT-SVD). Let SVD(X)= $U \Lambda V^T$. The SVD of $X L_R^T$, i.e. $U_R \Lambda_R V_R^T$, can be derived from SVD(X) as follows. Let V_R and Λ_R be the left Singular vector and Singular value of SVD($L_R V \Lambda$) whose right Singular matrix multiply by U will lead to U_R.

Proof: This can be converted into a Type-I LT-SVD problem by deriving SVD$(L_R X^T) = V_R \Lambda_R U_R^T$ from SVD(X^T). From Theorem 1, I first compute SVD$(L_R V \Lambda) = U_2 \Lambda_2 V_2^T$. Then I have $V_R = U_2$, $\Lambda_R = \Lambda_2$, and $U_R = U V_2$.

Example 2. (Bagging PCA) Suppose I have the same data matrix X as Example 1. Now I want to use bagging technique by selecting many subsets of size S to improve robustness of PCA basis for face recognition (Wang and Tang, 2006). Let e_i be a column vector whose i-th element is the only non-zero element with value 1. Let $L_k = [e_{k1}, ..., e_{ks}]$ be the sampling matrix for the k-th subset whose PCA is a type-II LT-SVD problem as SVD(XL_k).

Theorem 2. (Type-III LT-SVD) Given SVD$(X) = U\Lambda V^T$. Then SVD$(L_L X L_R^T)$ can be derived from SVD(X) as a Type-I LT-SVD followed by a Type-II LT-SVD.

Proof: From the Type-I LT-SVD, $SVD(L_L X) = U_L \Lambda_L V_L^T$ can be derived from SVD$(L_L U\Lambda)$. So $L_L X L_R^T = U_L \Lambda_L (L_R V_L)^T$ whose SVD is a Type-II LT-SVD in Corollary 1. I get the SVD$(\Lambda_L (L_R V_L)^T) = U_1 \Lambda_1 V_1^T$. So $L_L X L_R^T = (U_L U_1)\Lambda_1 V_1^T$. Since both U_L andU_1 are orthogonal matrix, and Λ_1 and V_1 are the SVD of $\Lambda_L (L_R V_L)^T$, I get SVD$(L_L X L_R^T) = U_{LR}\Lambda_{LR}V_{LR}^T$ where $U_{LR} = U_L U_1$, $\Lambda_{LR} = \Lambda_1$, and $V_{LR} = V_1$.

Example 3. (Rotation Forest) Rotation forest (Rodriguez *et al.*, 2006) is an ensemble classifier which builds a set of independent classifiers by two techniques to ensure their diversity. The first is bagging which randomly select subsets. The second is rotation which randomly split the feature into L subsets and run PCA on each feature subsets. From Example 2, bagging will produce a data matrix of XL_k, where L_k is the sampling matrix for k-th subset; From Example 1, the rotation will further produce a data matrix of $S_l X L_k$,where S_l is the sampling matrix for the l-th feature subset. The $SVD(S_l X L_k)$ can be derived from SVD(X) as Type-III LT-SVD with much less computations.

Since the SVD is related to the EVD of PSD matrix, the solution to the above three types of LT-SVD can be directly used for their peer LT-EVD problems.

Theorem 3. (Type-I LT-EVD) Given EVD(S)=$U\Sigma U^T$ for a PSD matrix S. Then EVD$(L_L S L_L^T) = U_L \Sigma_L U_L^T$ can be derived from SVD$(L_L U \Sigma^{1/2}) = U_1 \Lambda_1 V_1^T$ by $U_L = U_1$ and $\Sigma_L = \Lambda_1^2$. Let EVD(S)=$U\Sigma U^T$ where S is a PSD matrix. Then the Eigen vectors U_L and values Σ_L of EVD$(L_L S L_L^T)$ can be derived from SVD $(L_L U \Sigma^{1/2})$.

Proof: From EVD(S), $S = (U\Sigma^{1/2})(U\Sigma^{1/2})^T$. So $L_L S L_L^T = PP^T$ where $P = L_L U \Sigma^{1/2}$. Let SVD(P)=$U_L \Lambda_L V_L^T$ .Then $L_L S L_L^T = U_L \Lambda_L (V_L^T V_L) \Lambda_L U_L^T = U_L (\Lambda_L)^2 U_L^T$, which is exactly EVD(S) with Eigen vector U_L and Eigen value Λ_L^2.

Remark 3. (Truncated EVD) Similar to LT-SVD, I can use truncated EVD, i.e. $S \sim S_t = U_t \Sigma_t U_t^T$, for LT-EVD. From Theorem 3, I can solve SVD$(L_L U_t \Sigma_t^{1/2})$ for EVD$(L_L S_t L_L^T)$ to approximate EVD$(L_L S L_L^T)$ for the benefit on complexity. Let the size of L_L be $D \times N$. The complexity of EVD$(L_L S L_L^T)$ is $O(D^3)$ (Trefethen and Bau III, 1997), and that of SVD$(L_L U_t \Sigma_t^{1/2})$ is $O(t^2 D)$ if $t \ll D$ (which is usually the case). So the computation of LT-EVD is greatly reduced. How the approximation ($E = L_L(S - S_t)L_L^T$) affects Eigen vectors? Let u_i be the i-th Eigen vector of $L_L S L_L^T$ and u_i be that of $L_L S_t L_L^T$. From Theorem 8.1.2 in (Van Loan, 1996), I get $\langle u_i, \widetilde{u}_i \rangle = (1 - pp^T)^{-1/2}$ and$\|p\|_2 \leq (4\|E\|_2)/\delta_i$ with $\lambda_i(S)$ be the i-th Eigen value and $\delta_i = \min_{j \neq i} |\lambda_i(L_L S L_L^T) - \lambda_j(L_L S_t L_L^T)|$. So, similar to Remark 2, the error in Eigen value is controlled by EVD truncation error (E) and gap among Eigen values (δ_i).

Example 4. (Nystrom EVD) Let the rank-r kernel matrix K be $[K_{rr}, K_{nr}^T; K_{nr}, K_{nn}]$, where K_{rr} is the $r \times r$ sub-matrix of full rank. It implies $K = L_L K_{rr} L_L^T$ with $L_L = [I_{rr}; K_{nr} K_{rr}^{-1}]$. In Nystrom EVD problem ((Burges *et al.*, 2010)), I want to derive EVD(K) from EVD(K_{rr})= $U_r \Sigma_r U_r^T$, which is a Type-I LT-EVD problem. Nystrom EVD is widely used in manifold learning for out-of-sample extension, e.g. kernel PCA, and the usual solution is the Base-transform in Table 2.4, i.e.$U = L_L U_r = [U_r; K_{nr} U_r \Sigma_r^{-1}]$.

The following Lemma is very important before I can discuss the remaining two types of LT-EVD.

Lemma 1. The PSD matrix $S = XX^T$ has Eigen vectors of U and values of Σ if and only if U and $\Sigma^{1/2}$ are the left Singular matrix and Singular value of SVD(X).

Proof: The sufficient proof is to place SVD(X)=$U\Sigma^{1/2}V^T$ into $S = XX^T = U\Sigma U^T$, which is exactly EVD(S) because U is orthogonal and Σ is diagonal. The necessary proof is straight forward. Let $rank(X) = r$, then I have $XX^T = U_r \Sigma_r U_r^T$, where Σ_r consists of non-zero Eigen values with related Eigen vectors in U_r. Let $V_r = (\Sigma_r^{-1/2} U_r^T X)^T$. It is clear that $V_r^TV_r = I$. So SVD(X)=$U_r\Sigma_r^{1/2}V_r^T$.

Corollary 2. (Type-II LT-EVD) Given EVD(S)=U^T for a PSD matrix $S = XX^T$. Then EVD$(XL_R^TL_RX^T) = U_R\Sigma_RU_R^T$ can be derived from SVD$(\Sigma^{1/2}V^TL_R^T) = U_1\Lambda_1V_1^T$ by letting $U_R = UU_1$ and $\Sigma_R = \Lambda_1^2$.

Proof: Given EVD$(XX^T) = U\Sigma U^T$, from Lemma 1, SVD(X) is $U\Sigma^{1/2}V^T$ with $V = (\Sigma^{-1/2}U^TX)^T$. So $XL_R^TL_RX^T = UP_RP_R^TU^T$ where $P_R = \Sigma^{1/2}V^TL_R^T$. Let SVD($P_R$)=$U_1\Lambda_1V_1^T$. I have $XL_R^TL_RX^T = (UU_1)\Lambda_1^2(UU_1)^T = U_R\Sigma_RU_R^T$, where $U_R = UU_1$ and $\Sigma_R = \Lambda_1^2$. Since U and U_1 are column orthogonal, U_R is also orthogonal with $U_RU_R^T = I$. The Σ_R is diagonal because Λ_1 is the diagonal Singular matrix. So $XL_R^TL_RX^T = U_R\Sigma_RU_R^T$ is exactly the Eigen decomposition.

Corollary 3. (Type-III LT-EVD) Given EVD(S)=$U\Sigma U^T$ for a PSD matrix $S = XX^T$. Then the EVD of $L_LXL_R^TL_RX^TL_L^T$ can be directly derived from U and Σ.

Proof: Let $P_LR = L_LXL_R^T$ and SVD$(P_LR) = U_1\Lambda_1V_1^T$. From Lemma 1, the Eigen vector U_LR and Eigen value Σ_LR of $L_LXL_R^TL_RX^TL_L^T = P_LRP_LR^T$ is U_1 and Λ_1^2 respectively. So the problem remain is the derivation of SVD(P_LR)=SVD$(L_LXL_R^T)$ which is a Type-III LT-SVD problem if SVD(X) is known. Given EVD$(XX^T) = U\Sigma U^T$ and Lemma 1, I got SVD(X)=$U\Sigma^{1/2}V^T$ with $V = (\Sigma^{-1/2}U^TX)^T$. So the problem is solved.

Linear Transformed SVD for Tensor

In this section, I will extend the LT-SVD of matrix to its tensor case. An N-order tensor is an N-dimensional array noted as $\mathcal{X} \in R^{I_1\times I_2\times\ldots\times I_N}$. A tensor $\mathcal{X}$ can be unfolded into a matrix $X_{(n)} \in R^{I_n\times J}$, $J = \Pi_{i\neq n}I_i$ which takes the values along its n-th dimension as the column of $X_{(n)}$ and traverse the remaining dimensions. I can fold $X_{(n)}$ back to $\mathcal{X}$, and the inverse process is noted as $X_{(n)^{-1}}$. The n-Mode product is a multiplication between $\mathcal{X} \in R^{I_1\times I_2\times\ldots\times I_N}$ with a matrix $U \in R^{J\times I_n}$, and the result is a tensor of size

$R^{I_1 \times ... \times J \times ... \times I_N}$:

$$(\mathcal{X} \times_n U)_{i_1 ... j ... i_N} = \Sigma_{i_N=1}^{I_N} x_{i_1 ... i_n ... i_N} u_{j,i_n} \in R^{I_1 \times ... \times J \times ... \times I_N}. \tag{2.5}$$

The n-Mode product is the extension of matrix product which is connected by unfolded tensor:

$$\mathcal{Y} = \mathcal{X} \times_n U \Leftrightarrow Y_{(n)} = U \cdot X_{(n)} \tag{2.6}$$

The following properties (De Lathauwer *et al.*, 2000) can be easily derived from 2.5 and are important for latter discussions:

$$\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A (m \neq n). \tag{2.7}$$

$$\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA). \tag{2.8}$$

$$\mathcal{Y} = \mathcal{X} \times_1 A_1 \times_2 A_2 \times_N A_N \Leftrightarrow Y_{(n)} = A_n \cdot X_{(n)} (A_N \otimes ... A_{n+1} \otimes A_{n-1} ... \otimes A_1)^T \tag{2.9}$$

in which $\otimes$ denotes Kronecker product. I can also define SVD for tensors with many similarities to its matrix case. In fact, the High-order SVD (HOSVD) (De Lathauwer *et al.*, 2000) of a tenser $\mathcal{X} \in R^{I_1 \times \times I_n \times \times I_N}$ always exists as:

$$\mathcal{X} = \mathcal{S} \times_1 U_1 \times_2 U_2 \times_N U_N, \tag{2.10}$$

where the mode-n Singular matrix U_n is column-wise orthogonal and the core tensor $\mathcal{S} \in R^{I_1 \times \times I_n \times \times I_N}$ satisfies:

(i) All-orthogonality: Let the sub-tensor $\mathcal{S}_{i_n=\alpha}$ of $\mathcal{S}$ be obtained by fixing the n-th index to α. Then for every n and $\alpha \neq \beta$, I have $\langle \mathcal{S}_{i_n=\alpha}, \mathcal{S}_{i_n=\beta} \rangle = 0$, where$\langle \cdot, \cdot \rangle$ is the inner product by vectorizing the tensors in the same order .

(ii) Ordering: For every n, the Frobenius norm of sub-tensors follows $\|\mathcal{S}_{i_n=1}\|_F \geq \|\mathcal{S}_{i_n=2}\|_F \geq ... \geq \|\mathcal{S}_{i_n=I_n}\|_F$. The U_n in eq. 2.10 is comparable to the Singular matrix of matrix SVD, and $\mathcal{S}$ is comparable to Singular values. For 2-order tensor, the above

HOSVD is exactly the matrix SVD, and the core tensor $\mathcal{S}$ must be diagonal (De Lathauwer *et al.*, 2000).

It can be further shown that the decomposition in eq.2.10 is a HOSVD of $\mathcal{X}$ if and only if U_n is the left Singular matrix of $X_{(n)}(1 \leq n \leq N)$, and $\mathcal{S} = \mathcal{X} \times_1 U_1^T \times_2 ... \times_N U_N^T$. This leads to the HOSVD procedure in Fig.2.12. The HOSVD of $\mathcal{X}$ in eq.2.10 can be taken as a higher-order component analysis, and the eq.2.11 below shows the interaction between the core tensor and different component tenors:

$$\mathcal{X} = \Sigma_{i_1=1}^{I_1}\Sigma_{i_2=1}^{I_2}...\Sigma_{I_N=1}^{I_N} s_{i_1...i_N} \cdot (u_{i_1}^{(2)} \circ u_{i_2}^{(2)} \circ ... \circ u_{I_N}^{(n)}) \tag{2.11}$$

where $u_{i_k}^{(n)}$ is the i_k-th column of U_n, and $\circ$ is outer-product. Since $U_n(1 \leq n \leq N)$ are column-wise orthogonal, the component tensor $(u_{I_1}^{(1)} \circ u_{I_2}^{(2)} \circ ... \circ u_{I_N}^{(n)})$are orthogonal to each other. So eq.2.11 is an orthogonal decomposition of $\mathcal{X}$ with $\|\mathcal{X}\|_F^2 = \Sigma_{i_1=1}^{I_1}\Sigma_{i_2=1}^{I_2}...\Sigma_{i_N=1}^{I_N} s_{i_1...i_N}^2$. Truncated HOSVD takes the first $t_n(\leq I_n)$ vectors of U_n , i.e. $U_n^{(t_n)}$, and truncate S ,i.e. $\hat{\mathcal{S}} \in R^{t_1 \times ... \times t_N}$, to approximate $\mathcal{X}$:

$$\begin{aligned} \hat{\mathcal{X}} &= \hat{\mathcal{S}} \times_1 U_1^{(t_1)} \times_2 U_2^{(t_2)} \times_3 \ \times_N U_N^{(t_n)} \\ &= \Sigma_{i_1=1}^{t_1}...\Sigma_{i_N=1}^{t_N} s_{i_1...i_N} \cdot (u_{I_1}^{(1)} \circ u_{I_2}^{(2)} \circ ..._{I_N}^{(n)})). \end{aligned} \tag{2.12}$$

The approximation error is $\|\mathcal{X} - \hat{\mathcal{X}}\|_F^2 = \|\mathcal{S}\|_F^2 - \|\hat{\mathcal{S}}\|_F^2$ since U_n is orthogonal. In fact, HOSVD belongs to a more general class of dyadic decomposition, which has the same form of eq.2.10 with orthogonal U_n but free $\mathcal{S}$ from the constraints of all-orthogonality and ordering. The following two lemmas are necessary for the introduction of linear transformed HOSVD.

Lemma 2. Let SVD(M)=$U_m \Lambda_m V_m^T$. For any linear transform $W = UMV^T$ by column-wise orthogonal matrix U and V, I have SVD(W)=$(UU_m)\Lambda_m(VV_m)^T$, where UU_m and VV_m is the left and right Singular matrix respectively.

Proof. From SVD(M), I get $W = (UU_m)\Lambda_m(VV_m)^T$ which is the SVD of W since Λ_m is non-negative diagonal, and UU_m and VV_m are column-wise orthogonal.

Lemma 3. Let $u, v \in R^n$ satisfy $u \perp v$. Then for any $a, b \in R^m$, I have $u \otimes a \perp v \otimes b$.

Proof. Since $u \otimes a = [a_1 u^T, ..., a_m u^T]^T \in R^{n \times m}$ and $v \otimes b = [b_1 v^T, ..., b_m v^T]^T$, $\langle u \otimes a, v \otimes b \rangle = \Sigma_{i=1}^{m} a_i b_i \langle u, v \rangle = 0$.

Given any $\mathcal{Y} = \mathcal{X} \times_1 L_1 \times_2 ... \times_N L_N$, the linear transformed HOSVD (LT-HOSVD) problem is to derive HOSVD($\mathcal{Y}$) from HOSVD($\mathcal{X}$). Theorem 4 shows how I can achieve this.

Theorem 4. Given HOSVD($\mathcal{X}$)=$\mathcal{S} \times_1 U_1 \times_2 U_2 ... \times_N U_N$, the HOSVD of $\mathcal{Y} = \mathcal{X} \times_1 L_1 \times_2 ... \times_N L_N$ can be derived from that of $\mathcal{X}$ following the process in Fig.2.13.

Proof. Let $\mathcal{Y}^{(0)} = \mathcal{X}$ and $\mathcal{Y}^{(n)} = \mathcal{Y}^{(n-1)} \times_n L_n (1 \leq n \leq N)$. Then $\mathcal{Y} = \mathcal{Y}^{(n)}$. My proof consists of two parts w.r.t. the two stages in Fig.2.13. I first show that each loop in the forward stage produces a dyadic decomposition $\mathcal{Y}^{(n)} = \mathcal{H}^{(n)} \times_1 V_1^{(n)} \times_2 ... \times_N V_N^{(n)}$with $V_i^{(n)}$ be column-wise orthogonal, and $V_n^{(n)}$ is the mode-n Singular matrix of $\mathcal{Y}^{(n)}$. The proof is inductive. I start at $n = 0$. Since $\mathcal{H}^{(0)}$ and $V_n^{(0)}, (1 \leq n \leq N)$ is the HOSVD of $\mathcal{Y}^{(0)} = \mathcal{X}$, the above two conditions are satisfied. Now given $H^{(n-1)}$ and $V_i^{(n-1)}$ be the dyadic decomposition of $\mathcal{Y}^{(n-1)}$, I first show that the $\mathcal{H}^{(n)}$ and $V_i^{(n)} (1 \leq i \leq N)$ derived from line 3 to 5 is the dyadic decomposition of $\mathcal{Y}^{(n)} = \mathcal{Y}^{(n-1)} \times_n L_n$. In fact, from eq.2.7, eq.2.8 and eq.2.9, I have:

$$\begin{gathered} Y_{(n)}^{(n)} = L_n Y_{(n)}^{(n-1)} = (L_n V_n^{(n-1)}) \cdot \mathcal{H}_{(n)}^{(n-1)} \cdot \\ (V_N^{(n-1)} \otimes ... \otimes V_{n+1}^{(n-1)} \otimes V_{n-1}^{(n-1)} \otimes ... \otimes V_1^{(n-1)})^T \end{gathered} \tag{2.13}$$

Let SVD($L_n V_n^{(n-1)} \mathcal{H}_{(n)}^{(n-1)}$) $= U \Lambda V^T$. Then $Y_{(n)}^{(n)} = U(\Lambda V^T)(V_N^{(n-1)} \otimes ... \otimes V_{n+1}^{(n-1)} \otimes V_{n-1}^{(n-1)} \otimes ... \otimes V_1^{(n-1)})^T$. The $V_i^{(n)}$ and $\mathcal{H}^{(n)}$ updated in line 4 and 5 is dyadic decomposition of $\mathcal{Y}^{(n)}$, since U and $V_i^{(n-1)} (i \neq n)$ are column-wise orthogonal. Then, I show $V_n^{(n)} = U$ is the mode-n Singular matrix of $Y^{(n)}$, i.e. the left Singular matrix of $Y_{(n)}^{(n)}$. Since $V_i^{(n-1)} (1 \leq i \leq N)$ are orthogonal,their Kronecker in eq.2.13 is also orthogonal. From Lemma 2, the left Singular matrix of SVD($Y_{(n)}^{(n)}$) in eq.2.13 is that of SVD($L_n V_n^{(n-1)} \mathcal{H}_{(n)}^{(n-1)}$) which is exactly the U in line 3. The second part is to show that the $\mathcal{H}$ and $V_n (1 \leq n \leq N)$

generated in the backward stage is HOSVD($\mathcal{Y}$). For this purpose, I need to show that in each iteration of n, the updated $\mathcal{H}$ and V_n present a dyadic decomposition and V_n is the mode-n Singular matrix of $\mathcal{Y}$. The proof is inductive. The initiation in line 7 satisfies the two conditions. Suppose the conditions are still satisfied at n+1.I have $Y_{(n)} = V_n \mathcal{H}_{(n)}(V_N \otimes ... \otimes V_{(n+1)} \otimes V_{(n-1)} \otimes ... \otimes V_1)^T$ from eq.2.9. With similar augments in the forward stage based, the left Singular matrix of $Y_{(n)}$ is exactly that of $V_n \mathcal{H}_{(n)}$ (line 9). So the updated V_n is the mode-n Singular matrix, and the $\mathcal{H}$ (line 11) ensures that $\mathcal{H}$ and $V_n(1 \leq n \leq N)$ remain the dyadic decomposition of $\mathcal{Y}$. After iterations from N-1 to 1, I get all Singular matrices of $\mathcal{Y}$ and finished the HOSVD.

Remark 4. (Truncated LT-HOSVD). For $L_n \in R^{D_n \times I_n}$ and $\mathcal{X} \in R^{I_1 \times I_2 \times ... \times I_N}$, the complexity to compute HOSVD($\mathcal{Y}$) consists two parts. The first is to get the mode-n Singular matrix by SVD($Y_{(n)}$) from n=1 to N; the second is to get core tensor. Suppose $D_n \leq \Pi_{i \neq n} D_i$ for $1 \leq n \leq n$. It is easy to see the complexity of the two steps are both $O(\Pi_{n=1}^N D_n \cdot \Sigma_{n=1}^N D_n)$ which is the complexity of HOSVD($\mathcal{Y}$). I can use truncated HOSVD, i.e. the $\hat{\mathcal{X}}$ in eq.2.12, to approximate the HOSVD of $\mathcal{Y}$ that of $\hat{\mathcal{Y}} = \mathcal{X} \times_1 L_1 ... \times_N L_N$.The complexity of the truncated LT-HOSVD comes from the two stages of Fig.2.13. The major computation of the forward stage lies in the SVD in line 3, and that of the backward stage lies in the SVD in line 9. Suppose $D_n \leq \Pi_{i \neq n} t_i$, the complexity of each SVD is $O(D_n^2 \Pi_{i \neq n} t_i)$. So the total complexity is $O(\Pi_{n=1}^N t_n \cdot (\Sigma_{n=1}^N t_n^{-1} \cdot D_n^2))$, which is much less than that of direct HOSVD($\mathcal{Y}$) if $t_n \ll D_n$. The approximation error of the mode-n Singular vector (SVD($Y_{(n)}$)) is decided by both the Singular value gap and $\|Y_{(n)} - \hat{Y}_{(n)}\|_2)$ which is small given a small truncation error of $\hat{\mathcal{X}}$.

Remark 5. There are other types of LT-HOSVD which, in contrast to Theorem 4, only get linear transform in some modes $(L < N)$, i.e. $\mathcal{Y} = \mathcal{X} \times_{k_1} L_{k_1} ... \times_{k_L} L_{k_L}$. I can derive HOSVD($\mathcal{Y}$) from HOSVD($\mathcal{X}$) by some minor modification of the procedure in Fig.2.13. Specifically, in the forward stage (line 2 to 6), I only loop in mode-$k_i(1 \leq$

$i \leq L$). Following the proof of Theorem 4, the output is a dyadic decomposition of $\mathcal{Y}$ with V_{k_L} be the mode-k_L Singular matrix. Then, in the backward stage, I update other $V_n(1 \leq n \leq N, n \neq k_L)$ and $\mathcal{H}$ for the final HOSVD result. Since matrix is 2-order tensor, the procedure in Fig.2.13 is the procedure for Type-III LT-SVD in Theorem 2. Moreover, the traditional solutions to LT-SVD, e.g. re-SVD in Table 2.4, can be directly extended for LT-HOSVD in the same spirit.

2.3.3 Experiment

In this section, experiments are given to show the efficiency and accuracy of several solutions to LT-SVD and LT-HOSVD. The traditional three solutions (abbreviation) in Table 2.4, are compared with my solution based on SVD transform (ST). Since re-SVD is the precise solution of both LT-SVD and LT-HOSVD problems, I take its computation time and Singular vectors as the reference for comparison.

Efficiency

LT-SVD. I only compare the efficiency of RS with the ST, because BT and ID dont really compute SVD. Instead, the two methods transform either Singular vectors or the samples as quick approximation to LT-SVD. I only test the Type-I LT-SVD, because it is the base for other LT-SVD types.

The computation of RS lies in two parts. Let the size of Xbe $M \times N$ and L_L be $D \times M$. Then the complexity of the first part is multiplication, i.e. $L_L X$, with a complexity of $O(DMN)$; the second is the SVD of $L_L X$ with a complexity of $O(DN \cdot \min(D, N))$. I ONLY compare the complexity of the second part, because the SVD takes the major computation of re-SVD and this can free us from an extra parameter M in discussion. I fix the feature dimension D=2000, and test the sample size (N) on 3 different levels, i.e. 200 $(N \ll D)$, 2000 $(N \sim D)$, 20000 $(N \gg D)$. For each N, the first t Singular vectors are computed by our ST method defined in Theorem 1. I use random matrix and repeat 10 times for each pair of N and t, and the median time cost of both re-SVD (T_{RS}) and our ST (T_{ST}) are compared. The result in Fig.2.14 coincides with Remark 1. For large dataset ($N \gg D$), the computation of ST is much less than RS ($T_{ST} \sim 10^{-5} T_{RS}$ for $t = 5$) and the time cost will gradually increase with t. The time advantage of ST is not so obvious for small dataset (($N \ll D$)) . Note that if the first multiplication part of RS, i.e. $L_L X$, is included, the time gap between ST and RS will be further enlarged, because I only need to compute a much smaller multiplication, i.e. $L_L U_t \Lambda_t$, where multiplication by diagonal Λ_t is just a columnwise multiplication by scalar.

LT-HOSVD. I also compare the complexity of our ST to re-SVD. Let $\mathcal{Y} = \mathcal{X} \times_1 L_1 \times_2 L_2$ with $\mathcal{Y} \in R^{D \times D \times N}$. I fix D=100 and test 3 different data size (N), i.e. 2000 ($(N \ll D)^2$), 10000 ($N \sim D^2$), 50000 ($N \gg D^2$). In our ST, truncated HOSVD eq.2.12 with the first t Singular vectors on each dimension is used. I use random matrix and repeat 10 times for each pair of N and t and take the median time cost. As can be seen in Fig.2.15, the complexity of our ST is greatly reduced from re-SVD which coincides with Remark 4.

Accuracy

LT-SVD. The accuracy is illustrated in two aspects. The first is reconstruction error; the second is the similarity between the Singular vectors of re-SVD and that of other solution. I only test the Type-I LT-SVD, i.e. SVD($L_L X$), since it is the base of other LT-SVD types. In fact, the Singular vectors and their approximation error are decided by data matrix$X$and operator L_L based on which the SVD is computed. I use a data set of 10377 nature images with a uniform size of 128×128, and the L_L is a series of bilinear interpolation to resize the images into different scales. The i-th image is the i-th column in X.

The reconstruction error is the absolute error averaged over all sample dimensions and dataset, so it is independent to the size of sample and dataset. To be specific, let $\hat{U}_t$ be the first t approximated Singular vector of $L_L X$ provided by a LT-SVD solution. Then $\overline{\mathrm{err}} = (DN)^{-1} \cdot \|\widetilde{U}_t \widetilde{U}_t^T L_L X - L_L X\|_1$ is the reconstruction error, where $D \times N$ is the size of $L_L X$. Since ID approximately inverse $(\widetilde{L}_L^+)$ the test data into training domain and use the original SVD, i.e. $\hat{U}_t$, to compare with others, the error is achieved after re-transform into the testing domain, i.e. $\overline{\mathrm{err}} = (DN)^{-1} \cdot \|L_L(\widetilde{U}_t \widetilde{U}_t^T)(\widetilde{L}_L^+ L_L)X - L_L X\|_1$. In Fig.2.16, $\overline{\mathrm{err}}$ is divided by $(DN)^{-1} \cdot \|L_L X\|_1$, to be independent of value range, and two resize scales are tested: one is close to the original size (0.8), and the other is much smaller (0.4). It is clear that RS has the lowest reconstruction error, because it is the precise solution of LT-SVD. However, our ST is very close to RS, for it is a very good approximation (Remark 1). The error of ID and BT is much worse, because they are not precise approximation of SVD($L_L X$). In fact, the bases provided by BT are even not orthogonal to each other.

The gap between re-SVD and other methods increases with reconstruction dimension. In fact, their reconstruction errors are very close at the first several dimensions. The reason is that the first several Singular vectors derived from natural images are highly structured, and therefore easy to be approximated. But the structure will be gradually lost in subsequent Singular vectors. The reconstruction efficiency increase with resize scale. In fact, the average reconstruction error in scale 0.8 is much lower than that of scale 0.4 for all methods. The reason is similar to above: the image downsize will eliminate its inherent structure, and thus slower the SVD approximation.

I further investigate the similarity between the Singular vector of RS, i.e. u_0, and its provided by other solutions , i.e. u, via absolute cosine, i.e. $|\langle u, u_0 \rangle| \in [0, 1]$. The ID is not compared, for it does not compute Singular vectors explicitly.

Since the accuracy of our ST and BT are not on the same scale, I use $\log_{10}(1 - |\langle u, u_0 \rangle|)$ in Fig.2.17 where the average absolute cosine of first several percent of Singular values are compared ($2.5\% \sim 10\%$). For our ST, truncated SVD with first t vectors of$X$is used to approximate the SVD of $L_L X$. For BT, the accuracy is fixed, i.e. independent to t, because it always use L_L to transform the original Singular vectors of X. As I can see, the accuracy of ST is very high and will increase with more t for reconstruction. However, the accuracy of Singular vectors provided by ST will decrease for vectors related to smaller Singular values (compare the curve of 2.5% with that of 10% in Fig.2.17). The reason is that, for natural images, the leading Singular vectors are highly structured and enjoy a large gap among adjacent Singular values. From Remark 2, this implies a good approximation accuracy of Singular vectors. However, the structure and gap will gradually diminish.

LT-HOSVD. I use the same comparison measures and dataset as the above experiments for LT-SVD. Now the image dataset is organized as a tensor $\mathcal{X}$ where $\mathcal{X}(:,:,i)$ is the matrix related to the i-th image. The problem here is to get the HOSVD of $\mathcal{X} \times_1 L_1 \times_2 L_2$ where $L_n(n = 1, 2)$ is bilinear interpolation to resize the n-th dimension.

The comparison of reconstruction error in Fig.2.18 provides similar observations to that of LT-SVD in Fig.2.16.The precise Singular vectors by RS present the best approximation rate. Our ST is comparable to RS whose approximation error is much lower than that of ID and BT. The performance gap enlarges with either the increase of truncated dimension t (Singular vectors gradually lose their structure) or the increase of resize scale (downsize will make image smoother and easier to be approximated). However, there are two observations different than the LT-SVD. First, the approximation of RS, ST, and BT are very close at lower truncated dimension, which indicates the high similarity among their bases. Second, the approximation error of BT is even lower than that of ID in the case of low resize scale (0.4). The reason is that, in the tensor case, I resize the image in each dimension individually, i.e. $\mathcal{X} \times_1 L_1 \times_2 L_2$. As a result, the inverse transform adopted in ID is worse than direct bilinear interpolation of the whole image. The situation is severe for low resize scale.

The similarity between the Singular vectors of RS and the vectors provided by BT and our ST is provided in Fig.2.19, which shows similar result to those of LT-SVD. However, the angle between the vector provided by BT and Singular vector is much smaller than that of LT-SVD. The reason is that the Singular vector of HOSVD is derived individually for each image dimension. So it is very smooth and well approximated by bilinear interpolation.

2.3.4 Conclusion

In this chapter, I present the linear transformed SVD problem which arises when SVD or EVD is required by machine learning methods and testing domain can be characterized or approximated by a linear transform from training domain. The problem is quite general since SVD or EVD is essential to many machine learning methods, and a lot of transforms have their linear approximation as is illustrated by the examples in this chapter. In contrast to traditional solutions, our method achieves both efficiency and accuracy. I also extend the transformed SVD problem to its tensor case and provide solutions in the same spirit. My points on efficiency and accuracy are proofed by both theoretical analysis and numerical experiments. In future work, I will try to extend the transformed SVD problem to generalized Eigen problem.

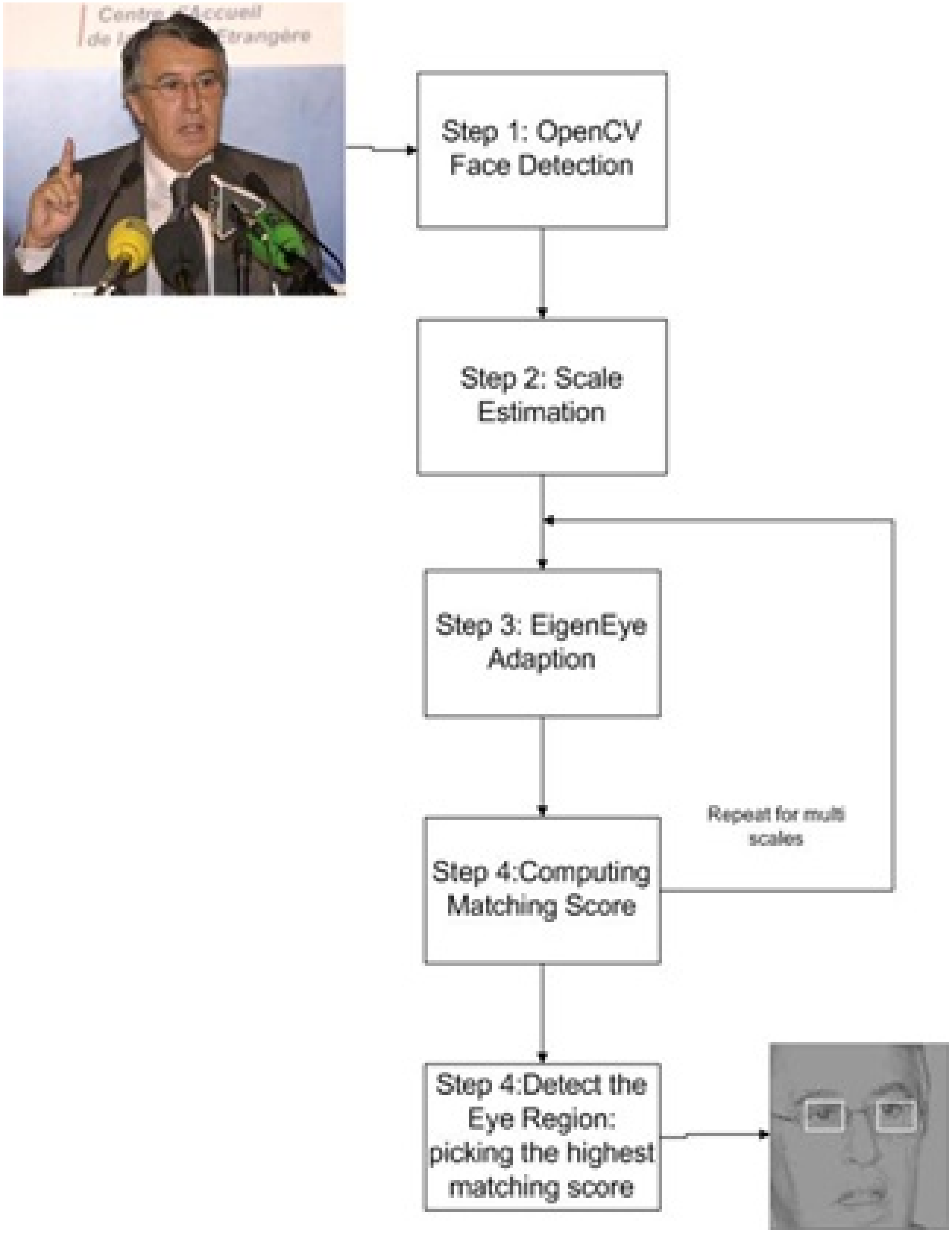

Centre d'Accueil
Step 1: OpenCV Face Detection
Step 2: Scale Estimation
Step 3: EigenEye Adaption
Step 4:Computing Matching Score
Repeat for multi scales
Step 4:Detect the Eye Region: picking the highest matching score

Figure 2.7: Complete Procedure.

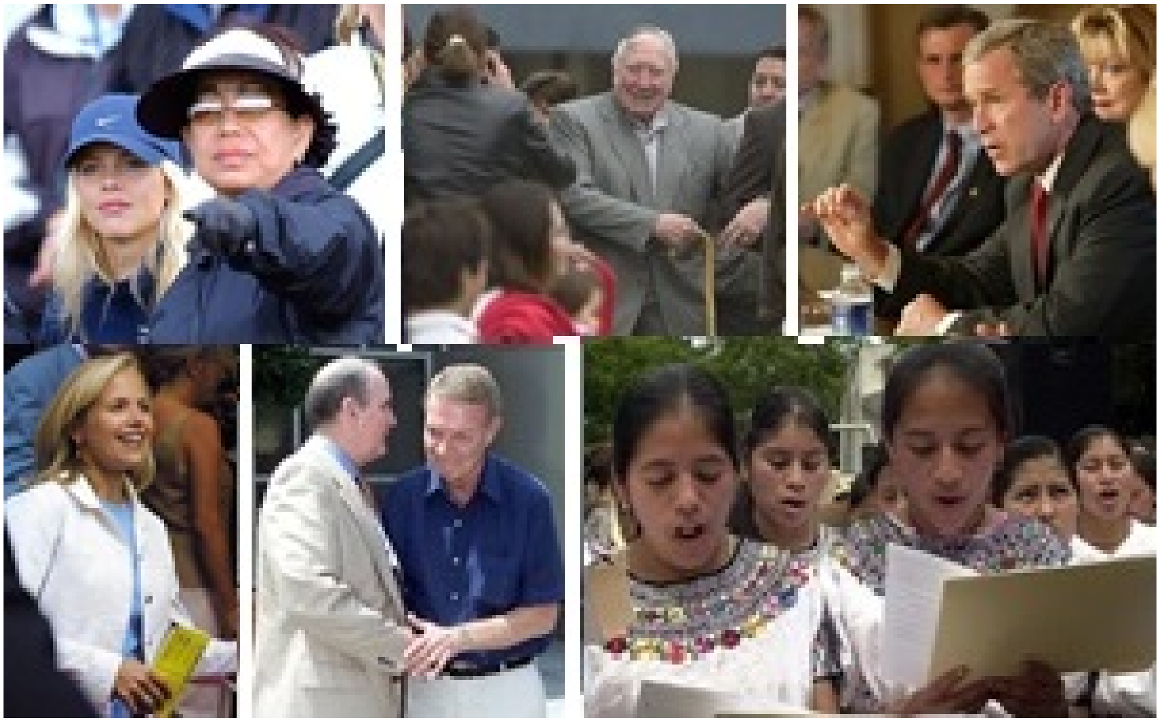


Figure 2.8: Sample Images from the FDDB Dataset

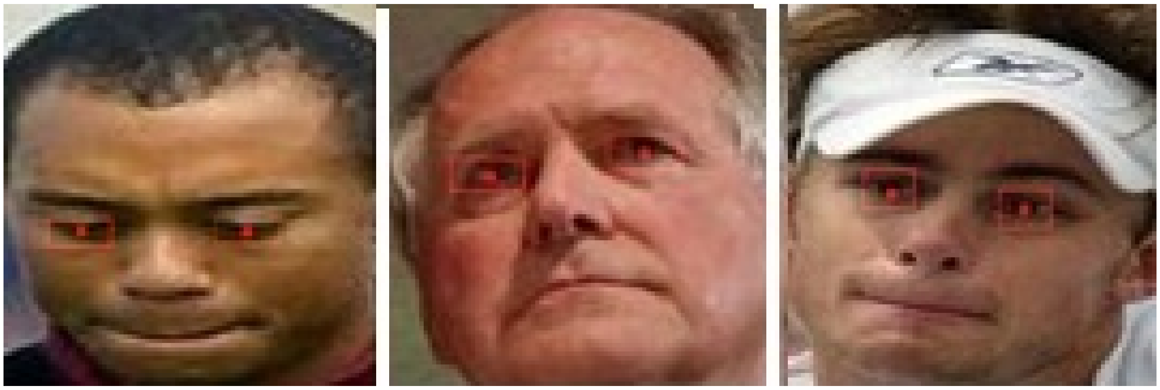


Figure 2.9: Illustrating the Manually-obtained Ground Truth.

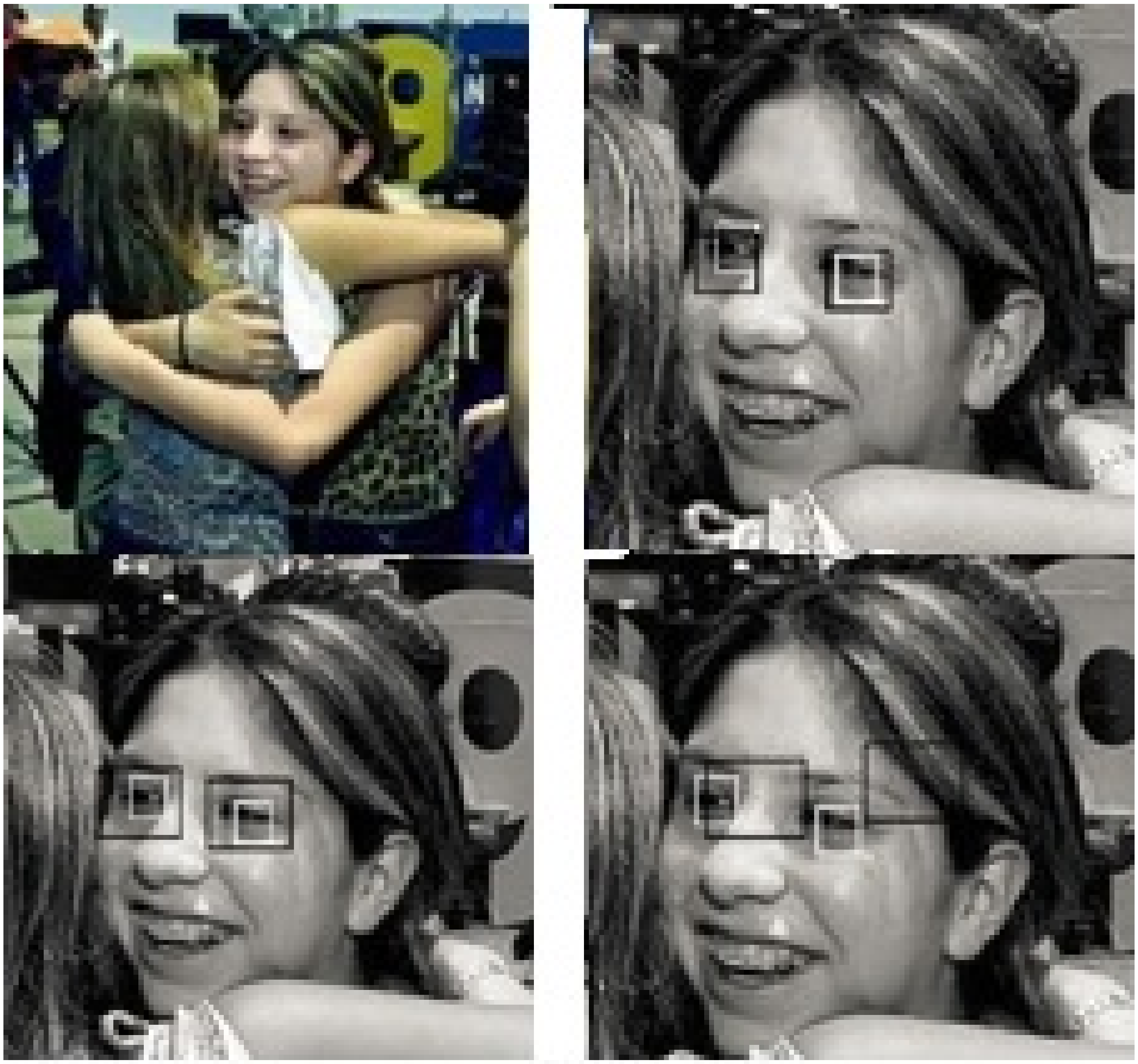


Figure 2.10: Original Emage; SAR eye Retection result;FVC Eye Detection Result; ICA Eye Detection Result.

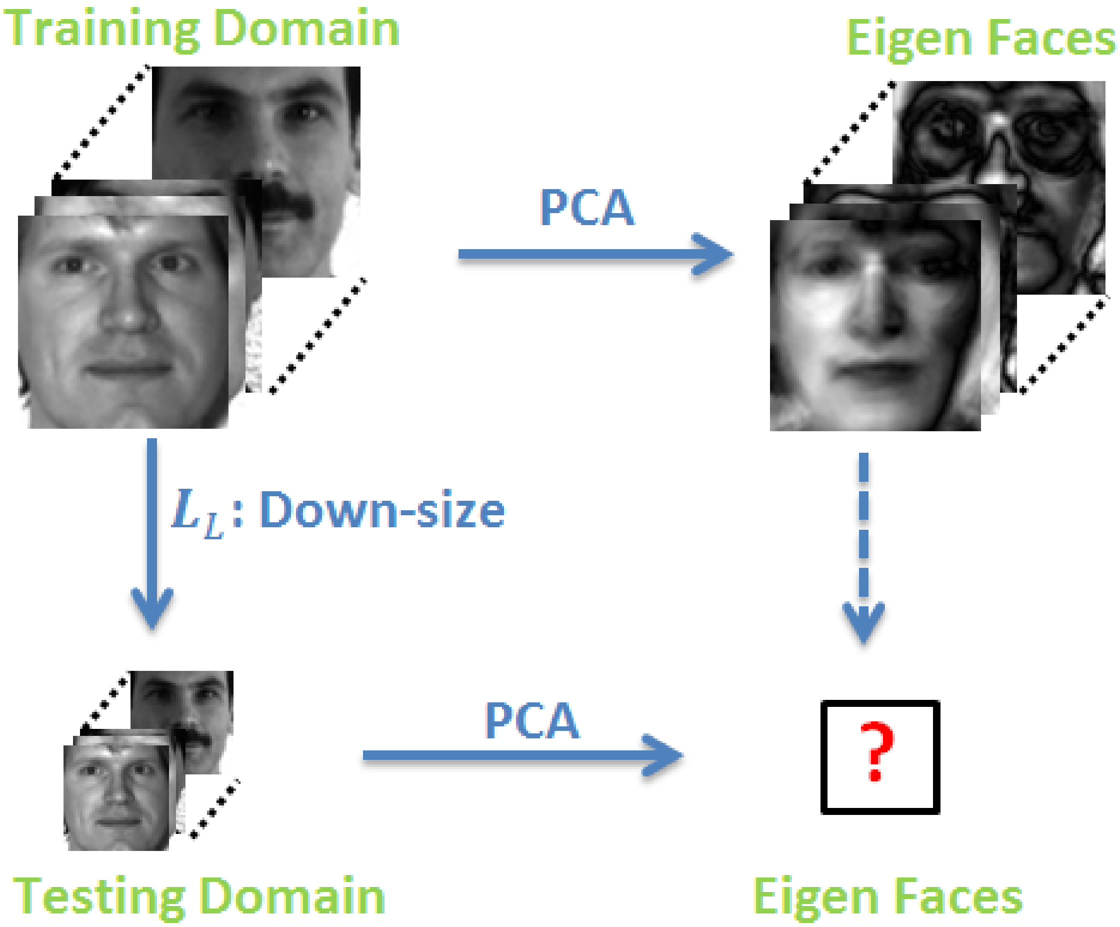
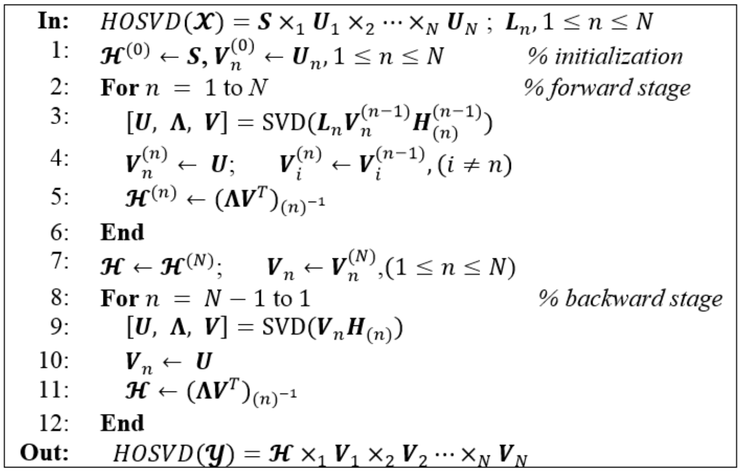


Figure 2.11: An Example of LT-SVD Problem as the PCA of Down-sized Face Images.

```
In:   $\mathcal{X} \in R^{I_1 \times I_2 \times \ldots \times I_N}$
1:    For $n = 1$ to $N$
2:        $[\boldsymbol{U}_n, \boldsymbol{\Lambda}_n, \boldsymbol{V}_n] = \mathrm{SVD}(\mathbf{X}_{(n)})$
3:    End
4:    $\boldsymbol{S} = \mathcal{X} \times_1 \boldsymbol{U}_1{}^T \times_2 \ldots \times_N \boldsymbol{U}_N{}^T$
Out:  $HOSVD(\mathcal{X}) = \boldsymbol{S} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_N \boldsymbol{U}_N$
```

Figure 2.12: Procedure of HOSVD for Tensor $\mathcal{X}$.



```
In:  $HOSVD(\mathcal{X}) = \mathcal{S} \times_1 \boldsymbol{U}_1 \times_2 \cdots \times_N \boldsymbol{U}_N$ ; $\boldsymbol{L}_n, 1 \le n \le N$
$\mathcal{H}^{(0)} \leftarrow \mathcal{S}, \boldsymbol{V}_n^{(0)} \leftarrow \boldsymbol{U}_n, 1 \le n \le N$          % initialization
For $n = 1$ to $N$                                  % forward stage
    $[\boldsymbol{U}, \boldsymbol{\Lambda}, \boldsymbol{V}] = \mathrm{SVD}(\boldsymbol{L}_n \boldsymbol{V}_n^{(n-1)} \boldsymbol{H}_{(n)}^{(n-1)})$
    $\boldsymbol{V}_n^{(n)} \leftarrow \boldsymbol{U}$;     $\boldsymbol{V}_i^{(n)} \leftarrow \boldsymbol{V}_i^{(n-1)}, (i \neq n)$
    $\mathcal{H}^{(n)} \leftarrow (\boldsymbol{\Lambda} \boldsymbol{V}^T)_{(n)^{-1}}$
End
$\mathcal{H} \leftarrow \mathcal{H}^{(N)}$;     $\boldsymbol{V}_n \leftarrow \boldsymbol{V}_n^{(N)}, (1 \le n \le N)$
For $n = N-1$ to $1$                                % backward stage
    $[\boldsymbol{U}, \boldsymbol{\Lambda}, \boldsymbol{V}] = \mathrm{SVD}(\boldsymbol{V}_n \boldsymbol{H}_{(n)})$
    $\boldsymbol{V}_n \leftarrow \boldsymbol{U}$
    $\mathcal{H} \leftarrow (\boldsymbol{\Lambda} \boldsymbol{V}^T)_{(n)^{-1}}$
End
Out:  $HOSVD(\mathcal{Y}) = \mathcal{H} \times_1 \boldsymbol{V}_1 \times_2 \boldsymbol{V}_2 \cdots \times_N \boldsymbol{V}_N$
```

Figure 2.13: Procedure of LT-HOSVD for $Y = X \times_1 L_1 \times_2 ... \times_N L_N$.

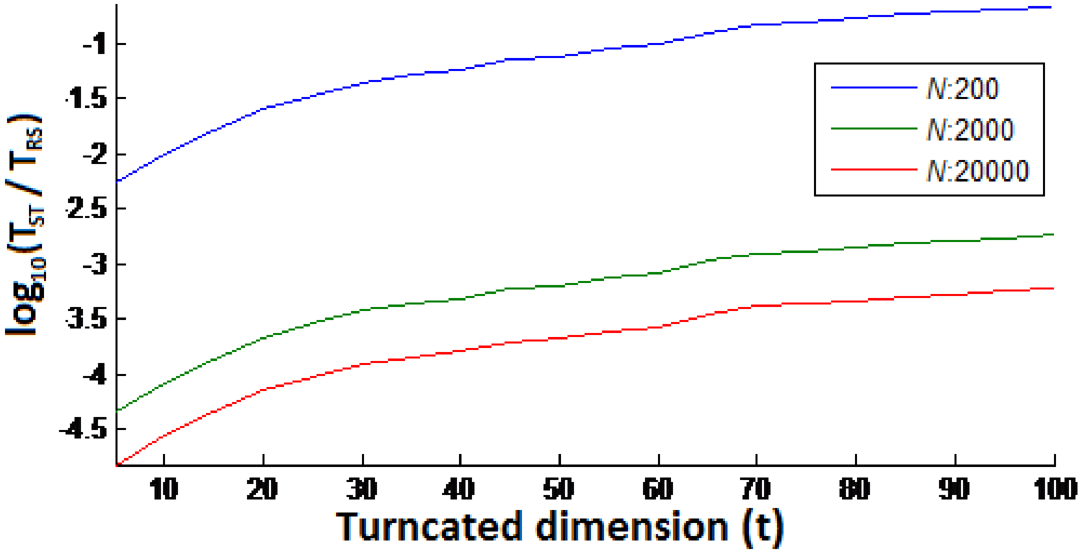


Figure 2.14: Comparison of Complexity Between Re-SVD (T_{RS}) and T_{ST} for Type-I LT-SVD, with $L_L X \in R^{D \times N}$ and D=2000.

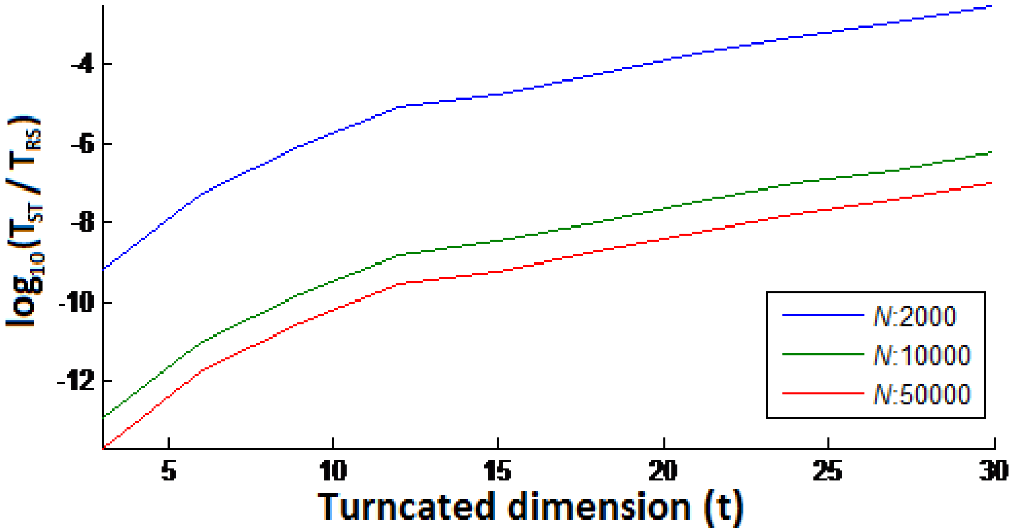


Figure 2.15: Comparison of Complexity Between Re-SVD (T_{RS}) and T_{ST} for LT-HOSVD of $\mathcal{Y} = \mathcal{X} \times_1 L_1 \times_2 L_2$ with $\mathcal{Y} \in R^{D \times D \times N}$ and D=100.

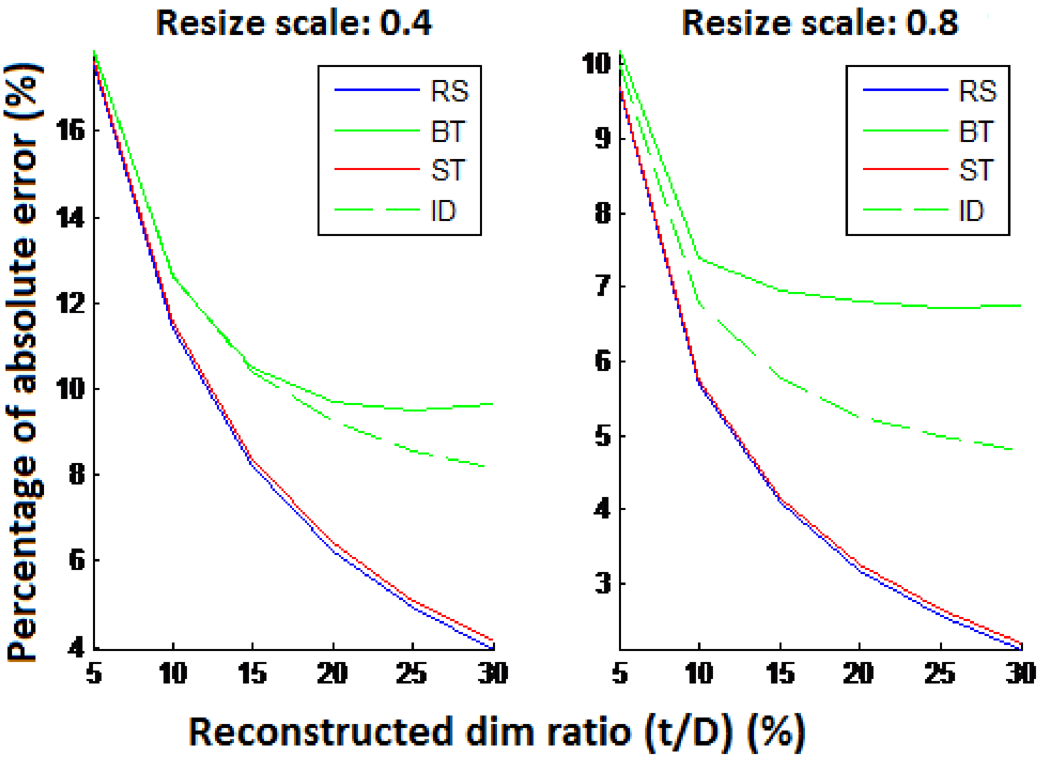


Figure 2.16: Comparison of Reconstruction Error for Type-I LT-SVD between ST and Other Solutions.

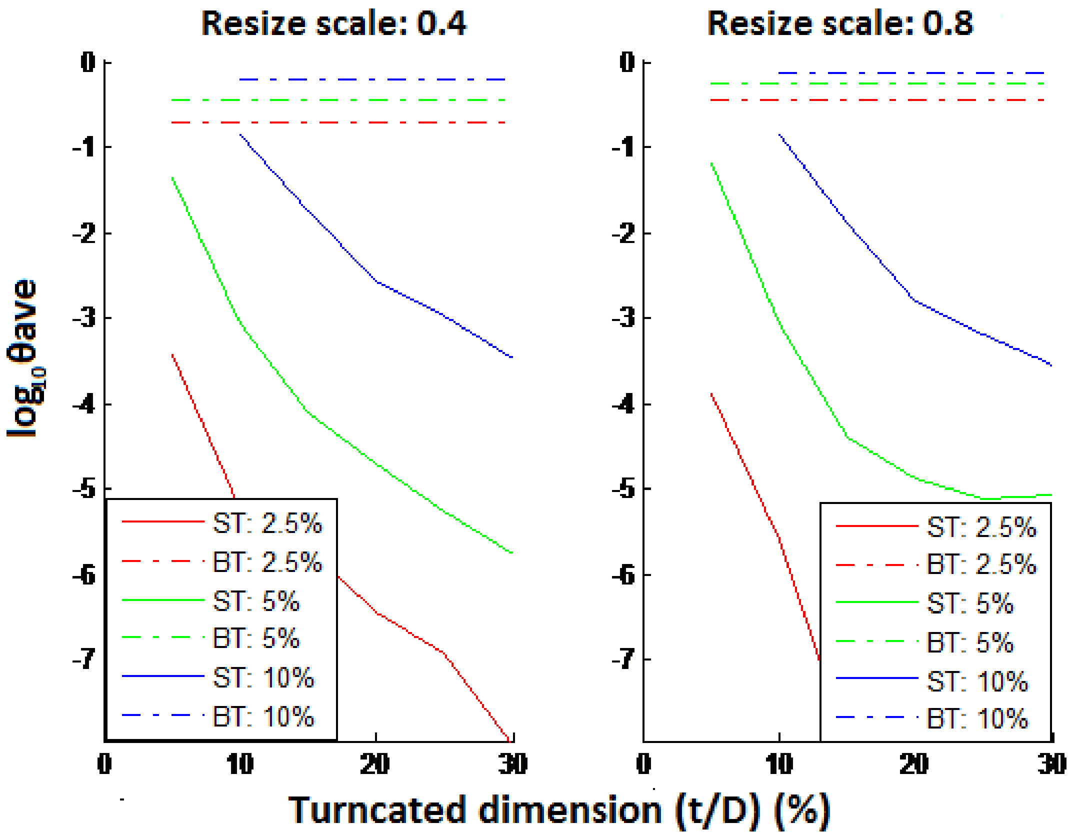


Figure 2.17: Accuracy Comparison of LT-SVD by the Average Inner Product (θ_{ave}) between the First ($2.5\% \sim 10\%$) Singular Vectors and Those Provided by ST and BT.

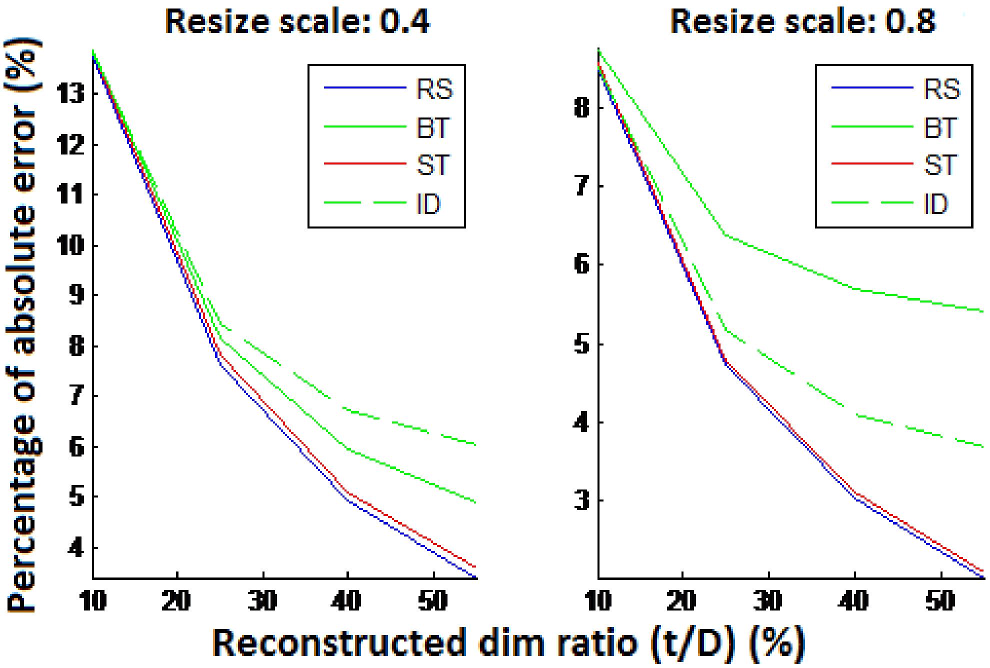


Figure 2.18: Comparison of Reconstruction Error by Different LT-HOSVD Methods.

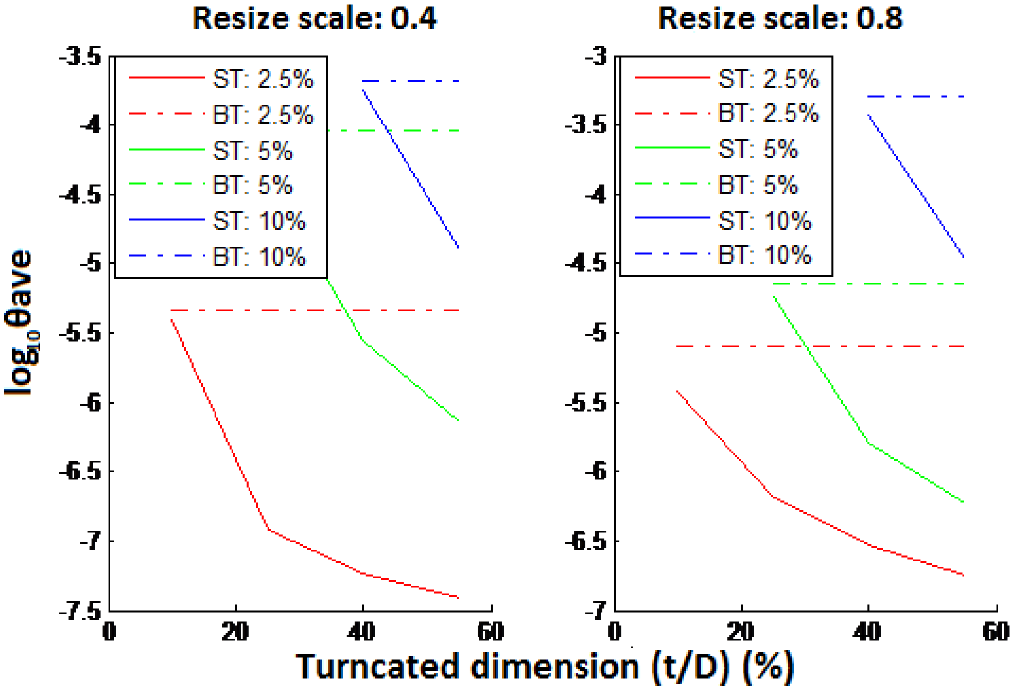


Figure 2.19: Accuracy Comparison of LT-HOSVD by the Average Inner Product (θ_{ave}) between the First ($2.5\% \sim 10\%$) Singular Vectors and Those Provided by ST and BT.

Chapter 3

LEARNING WITH PARTIALLY LABELED DATA AND NOISY DATA

3.1 Non-negative Dictionary Learning with Pairwise Partial Similarity Constraint

3.1.1 Introduction

Face retrieval and recognition has been a popular topic and is intensively studied in recent years. Due to the increasing number of available multimedia sources,fast and robust face retrieval remains a challenge task. Sparse coding based on over-completed dictionaries has been widely used in many applications in visual computing including face recognition and retrieval. Although a baseline dictionary D could be formed directly from the training data samples (as in Wright *et al.* (2009)), learning a proper dictionary D is often a central task of such approaches. A compact dictionary may be learned by the K-SVD algorithm Aharon *et al.* (2005) or its variants. For making the learned dictionary more effective for classification (in addition to representation), discriminative dictionary-learning algorithms have also been proposed, such as the D-KSVD algorithm Zhang and Li (2010) and the LC-KSVD algorithm Jiang *et al.* (2011). Other examples of this sort include the methods proposed in Yang and Zhang (2010), Shenghua Gao (2010), Yang *et al.* (2011a), Quach *et al.* (2014), Theodorakopoulos *et al.* (2011), Shiau and Chen (2012) for face recognition and other classification tasks.

The above approaches for increasing the discriminative power of the learned dictionary rely on the class labels of the training samples.But in real applications, massive multimedia data often does not have complete labels. Many types of methods have also been proposed for applications that lack such full labeling information. For example, pairwise matching labels (indicating whether each pair of training sample is the same or different) were used in Guo *et al.* (2013) for learning a discriminative dictionary. The latter approach, greatly improve the quality of the dictionary under such limited label information. While the discriminative power of a dictionary often comes from the class label information, Guo *et al.* (2013) provides the discriminative power from only the pairwise label('same' and 'different'). In this chapter, I propose a novel framework to learn a dictionary when even fewer label information is available - only the 'same' labels are available. To be specific, given two signals, if the label is 'same', they are from the same class, otherwise they may or may not come from the same class.

In this chapter I deal with a similar problem but with much less labeling information: the training data contain only limited sample pairs that should be labeled as the same. Further, I allow the matched pairs to be similar only on a portion of the data samples. Such a problem naturally arises in real applications like face retrieval, where the training set may contain several labeled instances of some subjects and the goal is to learn an effective dictionary for retrieving faces. And clearly in order to accommodate localized differences among images of the same subject (e.g., due to occlusion by sun glasses or variations of facial expression), I will need to confine the pairwise similarity constraint to only a portion of the data samples (with the location possibly varying from one pair to another). How to learn a discriminative dictionary under such constraints is a new problem that cannot be addressed by the prior work.

I propose a learning framework for addressing the above problem. my formulation of the problem supports incorporation of limited pairwise label information as well as representation of the rest of the data. Further, the approach allows the pairwise partial similarity constraint to be imposed on the input pairs in a sample-adaptive manner. This is partially achieved first by imposing the non-negativity constraint on sparse coding (similar to non-negative matrix factorization for face recognition Guillamet and Vitria (2002), Wang *et al.* (2005b), Zhang *et al.* (2008), which naturally leads to a part-based representation, but more importantly by the introduction of a “selection operator” that explicitly specifies which parts of the input samples are considered for imposing the similarity constraint. I design an optimization algorithm for finding the optimal solution under the proposed framework, and I prove its convergence. I show that even with the limited label information, a discriminative dictionary can still be learned. The learned dictionary enforce signals from the same class to have similar sparse representation. My dictionary learning approach also explicitly impose the non-negativity constraint to the sparse code and dictionary, and a pairwise partial similarity constraint. The pairwise partial similarity constraint, force partial elements of the pair of signals from the same class to be close. For a pair of signals like images of human face, even if they are from the same class, the distance between them can be very large due to distortion and occlusion. Thus the pairwise partial similarity constraint is robust to these distortion. I report results from experiments designed to systematically evaluate the proposed method.

The rest of the chapter is organized as follows. I first formulate the problem, and then present an optimization algorithm for finding a solution under my formulation. Furthermore, I provide the proof of convergence of the proposed algorithm. Experimental results are reported next, which are followed by conclusion.

3.1.2 Formulating the Problem

For completeness, I first describe the basic dictionary learning problem. Let $X = [x_1, ..., x_N] \in \mathbb{R}^{M \times N}$ be a set of N M-dimensional input vectors. Learning a dictionary for reconstruction with K atoms for the sparse representation of X can be written as the following minimization problem:

$$< D^*, A^* >=_{D,A} \|X - DA\|_F^2 + \lambda \|A\|_{1,1} \tag{3.1}$$

where $D = [d_1, ..., d_K] \in \mathbb{R}^{M \times K}$ ($K > M$, making the dictionary over-complete) is the learned dictionary, $A = [\alpha_1, ..., \alpha_N] \in \mathbb{R}^{K \times N}$ is the sparse representations of the input vectors in X, $\|X - DA\|_F^2$ represents the reconstruction error, λ is the sparsity constraint factor. This problem can be further split into the following two sub-problems (3.2) and (3.3):

$$D^* =_D \|X - DA\|_F^2 \tag{3.2}$$

$$A^* =_A \|X - DA\|_F^2 + \lambda \|A\|_{1,1} \tag{3.3}$$

The construction of D can be achieved by solving Equation (3.2). One common approach, the K-SVD algorithm Aharon *et al.* (2006), has been used widely in many applications. When D is given, the task of sparse coding is to compute the sparse representation A of X by solving (3.3).

Now, considering the desired properties for the new dictionary learning problem discussed in the previous section, I propose the following dictionary learning problem:

$$\begin{aligned} < D, A > &=_{D,A} \|X - DA\|_F^2 + \lambda \|A\|_{1,1} \\ &+ \frac{\mu}{2} \sum_{i,j} \|P^{ij} D(\alpha_i - \alpha_j)\|_2^2 \\ &=_{D,A} \|X - DA\|_F^2 + \lambda \|A\|_{1,1} \\ &+ \frac{\mu}{2} \sum_{i,j} \|P^{ij} DA(J^{i1} - J^{j1})\|_F^2 \\ &s.t. \quad D \geq 0, A \geq 0 \end{aligned} \tag{3.4}$$

Compared with the basic problem of (1), there are some differences in the definitions of the symbols, which are explained below. $X = [x_1, x_2, ..., x_N] \in \mathbb{R}_+^{M \times N}$ denotes the training data with N signals, $D = [d_1, d_2, ..., d_K] \in \mathbb{R}_+^{M \times K}$ is the dictionary to be learned, $A = [\alpha_1, \alpha_2, ..., \alpha_N] \in \mathbb{R}_+^{K \times N}$ is the sparse representation of input data X, and $\mathbb{R}_+ = [0, \infty)$ is the set of non-negative real number. J^{ij} is a single-entry matrix, where its element equals to 1 at (i, j), 0 elsewhere. P^{ij} is a diagonal matrix which choose the largest $\varphi\%$ elements from $D(\alpha_i - \alpha_j)$ if α_i and α_j are from the same class. If they are from different class, P^{ij} equals to a zero matrix. To be precise for the first case, if $(D(\alpha_i - \alpha_j))_n$, the nth element of $D(\alpha_i - \alpha_j)$, is among the largest $\varphi\%$ elements of $D(\alpha_i - \alpha_j)$, I set $(P^{ij})_{nn} = 1$ else $(P^{ij})_{nn} = 0$.

In the objective function, $\|X - DA\|_F^2$ represents the reconstruction error, $\|A\|_{1,1}$ represents the sparsity regularization, and $\sum_{i,j} \|P^{ij} D(\alpha_i - \alpha_j)\|_2^2$ is the pairwise partial similarity constraint, which requires that only the chosen percentage of pixels of the reconstructed image are needed to be close, if the two images are from the same object. The parameters λ and μ are the weights of the sparsity regularization and pairwise partial similarity constraint respectively.

In the above formulation, the constraint of pairwise partial similarity has been imposed through the term $\sum_{i,j} \|P^{ij} D(\alpha_i - \alpha_j)\|_2^2$. Here, P serves as a sample-adaptive selector such that only a chosen percentage of pixels is included in comparison of two images. Note that my model does not necessary required any fixed particular components of the images to be chosen since P is sample-adaptive.

An Algorithm Aolving the Optimization

In this section, I present an algorithm for solving the optimization problem defined in the previous section. It is challenging to solve (3.4) since it is not convex in both D and A together simultaneously. Fortunately, the problem is convex when one of the variables is fixed. Hence I solve (3.4) by an iterative multiplication update rule, updating the variables one at a time as follows.

Update D: when A is fixed, only the second term is independent of D, so the equation of updating D can be written as:

$$\min_{D} ||X - DA||_F^2 + \frac{\mu}{2}\sum_{i,j} \|P^{ij}D(\alpha_i - \alpha_j)\|_F^2 \quad s.t. \quad D \geq 0 \tag{3.5}$$

The update rule for D can be written as:

$$D_{mn} \leftarrow D_{mn}\frac{(XA^T + \mu\sum_{i,j} P^{ij}D(\alpha_i\alpha_j^T + \alpha_j\alpha_i^T))_{mn}}{(DAA^T + \mu\sum_{i,j} P^{ij}D(\alpha_i\alpha_i^T + \alpha_j\alpha_j^T))_{mn}} \tag{3.6}$$

Update A: To update A, I fix D and solve the following problem:

$$\min_{A} \|X - DA\|_F^2 + \lambda\|A\|_{1,1} + \frac{\mu}{2}\sum_{i,j} \|P^{ij}D(\alpha_i - \alpha_j)\|_2^2 \tag{3.7}$$

By introducing a new variable γ, (3.7) can be split into two minimization problem (3.8) and (3.9):

$$\begin{aligned} \min_{A} &\|X - DA\|_F^2 + \lambda'\|\gamma - A\|_F^2 \\ &+\frac{\mu}{2}\sum_{i,j} \|P^{ij}D(\alpha_i - \alpha_j)\|_2^2 \quad s.t. \quad A \geq 0 \end{aligned} \tag{3.8}$$

and

$$\min_{\gamma} \lambda\|\gamma\|_{1,1} + \lambda'\|\gamma - A\|_F^2 \quad s.t. \quad \gamma \geq 0 \tag{3.9}$$

The following update rule can be used to solve (3.8):

$$A_{mn} \leftarrow A_{mn}\frac{(D^TX + \lambda'\gamma + \mu\sum_{i,j} D^TP^{ij}DAJ^{ij})_{mn}}{(D^TDA + \lambda'A + \mu\sum_{i,j} D^TP^{ij}DAJ^{ii})_{mn}} \tag{3.10}$$

The solution of (3.9) is given in Goldstein and Osher (2009), by using the shrinkage operator:

$$\gamma_{mn} = shrink(A_{mn}, \frac{\lambda}{2\lambda'}), \tag{3.11}$$

where

$$shrink(a, b) = \frac{a}{|a|} * \max(|a| - b, 0) \tag{3.12}$$

It is clear that γ will be non-negative if it is initialized with non-negative values. The convergence of (3.9) under (3.11) has been proved in Goldstein and Osher (2009). Algorithm 1 **??** summarizes the overall computing steps of the proposed optimization algorithm.

Algorithm 1 The proposed algorithm

```
Input: $X$,$D_0$,$A_0$,$\gamma_0$,parameters: $\lambda, \lambda', \mu$
Output:$D$,$A$
1: While Not convergent or $i <=$ max iteration number do
2:     Update $D$ using Eq.3.6;
3:     Update $A$ using Eq.3.10;
4:     Update $\gamma$ using Eq.3.11;
```

3.1.3 Convergence of the Proposed Algorithm

I now prove the convergence of my algorithm. The proofs are based on either existing results of Lee and Seung (2001) or my extension of some results therein to a more general situation matching my problem formulation. Some basic definitions are given first. A function $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is an auxiliary function for $f : \mathbb{R}^n \to \mathbb{R}$ if

$$\forall x, y \in \mathbb{R}^n, \quad g(x, y) \geq f(x) \quad and \quad g(x, x) = f(x) \tag{3.13}$$

Let $T : \mathbb{R}^n \to \mathbb{R}^n$ be an operator. A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be decreasing (or non-increasing) under the update rule $y^{(t+1)} = T(y^{(t)})$ if

$$f(y^{(t+1)}) \leq f(y^{(t)}), \quad \forall t \in \{0, 1, 2, \ldots\} \tag{3.14}$$

Lemma 1. *Lee and Seung (2001) Let $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be an auxiliary function for $f : \mathbb{R}^n \to \mathbb{R}$. Then f is decreasing under the update rule*

$$y^{(t+1)} =_y g(y, y^{(t)}) \tag{3.15}$$

Lemma 2. *Let $y \in \mathbb{R}_+^n$, $f : \mathbb{R}_+^n \to \mathbb{R}_+$ be a function such that $\mathbf{H}f(y) \in \mathbb{R}_+^{n\times n}$. Define $K : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ as follow:*

$$K_{ij}(y) = \left(\mathbf{H}f(y)y\right)_i \delta_{ij}/y_i \tag{3.16}$$

where $\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ if $i \neq j$. Then $K(y)$ is a diagonal matrix. Also, if $y, y^{(k)} \in \mathbb{R}_+^n$, then

$$\begin{aligned} g(y, y^{(t)}) = f(y^{(t)}) + (y - y^{(t)})^T \nabla f(y^{(t)}) \\ + \frac{1}{2}(y - y^{(t)})^T K(y^{(t)})(y - y^{(t)}) \end{aligned} \tag{3.17}$$

is an auxiliary function for $f(y)$.

Proof. $g\left(y, y\right) = f(y)$ is trivial. To prove $g\left(y, y^{(t)}\right) \geq f(y)$, I first expand f:

$$\begin{aligned} f(y) = f(y^{(t)}) + (y - y^{(t)}) \nabla f(y^{(t)}) \\ + \frac{1}{2}(y - y^{(t)})^T \mathbf{H}f(y^{(t)})(y - y^{(t)}) \end{aligned} \tag{3.18}$$

Then,

$$\begin{aligned} & g(y, y^{(t)}) \geq f(y) \\ \Leftrightarrow & \left(y - y^{(t)}\right)^T (K(y^{(t)}) - \mathbf{H}f(y^{(t)}))(y - y^{(t)}) \geq 0 \end{aligned} \tag{3.19}$$

To prove $K(y^{(t)}) - \mathbf{H}f(y^{(t)})$ is positive semidefinite. Consider

$$B_{pq}(y^{(t)}) = \left(y^{(t)}\right)_p \left(K(y^{(t)}) - \mathbf{H}f(y^{(t)})\right)_{pq} \left(y^{(t)}\right)_q \tag{3.20}$$

which rescales the components of $K - \mathbf{H}f$. So, $K - \mathbf{H}f$ is positive semidefinite $\Leftrightarrow$ B is positive semidefinite.

$\forall v \in \mathbb{R}^n$,

$$\begin{aligned}
v^T B v =& \sum_{pq} v_p B_{pq} v_q && (3.21)\\
=& \sum_{pq} \Big[(y^{(t)})_p (\mathbf{H} f(y^{(t)}))_{pq} (y^{(t)})_q {v_p}^2 && (3.22)\\
& - v_p (y^{(t)})_p (\mathbf{H} f(y^{(t)}))_{pq} (y^{(t)})_q v_q \Big] && (3.23)\\
=& \sum_{pq} (\mathbf{H} f(y^{(t)}))_{pq} (y^{(t)})_p (y^{(t)})_q && (3.24)\\
& (\frac{1}{2}{v_p}^2 + \frac{1}{2}{v_q}^2 - v_p v_q) && (3.25)\\
=& \frac{1}{2} \sum_{pq} (\mathbf{H} f(y^{(t)}))_{pq} (y^{(t)})_p (y^{(t)})_q (v_p - v_q)^2) && (3.26)
\end{aligned}$$

So, B is positive semidefinite $\Rightarrow g\left(y, y^{(t)}\right) \geq f\left(y\right)$ □

Combining (3.15) in Lemma 1 and (3.17) in Lemma 2, I can conclude that under the update rule

$$y^{(t+1)} = y^{(t)} - K(y^{(t)})^{-1} \nabla f(y^{(t)}) \tag{3.27}$$

f is decreasing.

The objective function in (3.8) is decreasing under the update rule in (3.10).

Proof. If I update α_i one by one, and fix other α_j ($j \neq i$), consider

$$\begin{aligned}
f(\alpha_i) = \|x_i - D\alpha_i\|_2^2 + \lambda' \|\gamma_i - \alpha_i\|_2^2 \\
+ \frac{\mu}{2} \sum_j \|P^{ij} D(\alpha_i - \alpha_j)\|_2^2
\end{aligned} \tag{3.28}$$

Then (3.8) is equivalent to $\min_{\alpha_i} f(\alpha_i)$. I then calculate the gradient and Hessian of f:

$$\begin{aligned}
\nabla f(\alpha_i) = (2D^T D + 2\lambda' I + 2\mu \sum_{j \neq i} D^T P^{ij} D)\alpha_i \\
-(2D^T x_i + 2\lambda' \gamma_i + 2\mu \sum_{j \neq i} D^T P^{ij} D\alpha_j)
\end{aligned} \tag{3.29}$$

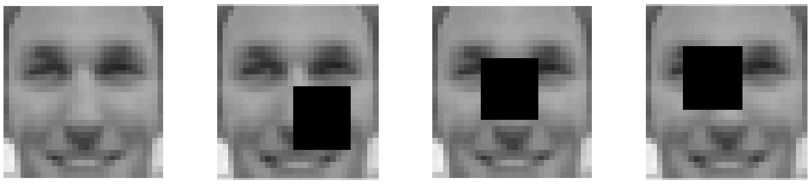


Figure 3.1: Sample Images Used in Experiment 1.

$$\mathbf{H}f(\alpha_i) = 2D^T D + 2\lambda' I + 2\mu \sum_{j \neq i} D^T P^{ij} D \tag{3.30}$$

The proof is achieved by substituting (3.29) and (3.30) into (3.27). □

The objective function in (3.5) is decreasing under the update rule in (3.6).

Proof. The proof is similar to the one for the update rule in (3.8): First define a function of d_i, then find its gradient and Hessian, and substitude it into (3.27). □

Also, the objective function in (3.9) is decreasing under the update rule in (3.11) Goldstein and Osher (2009). Hence by the above proofs, the objective function in (3.4) converges under the proposed algorithm.

3.1.4 Experiments and Results

I conduct three experiments: the first one uses simulated occlusion to face images to verify the idea and the second one uses a subset of AR face dataset Martinez (1998) of challenging face images and demonstrate face retrieval task using my proposed method. The third experiment uses a subset of both AR and LFW dataset to demonstrate face verification task. All experiments show the superiority of the dictionary learned by my proposed method.

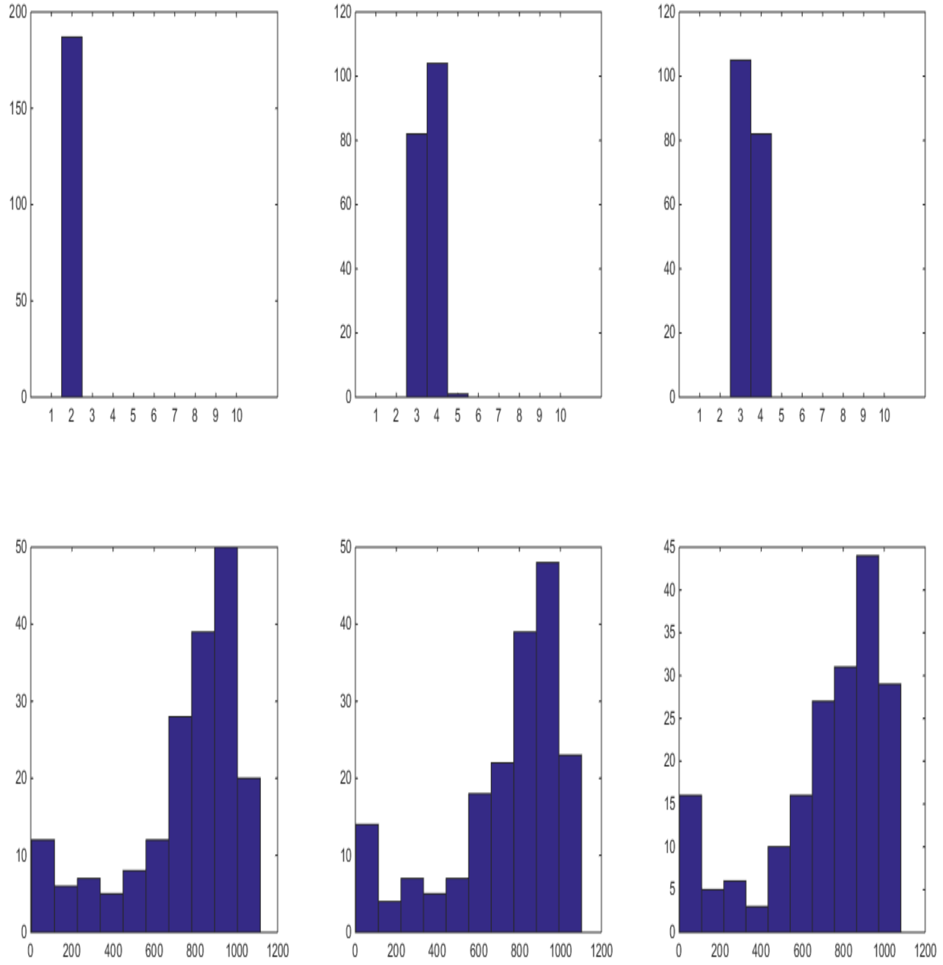


Figure 3.2: Histogram of the Rankings.

Experiment 1: Evaluation with Simulated Occlusion

The purpose of the first experiment is to verify the correctness of my proposed method and its optimization procedure. Ergo I use synthetic data with a simple experimental setup. I use a frontalized version of LFW dataset Hassner *et al.* (2015), choosing 187 images from 187 subjects. Then I overlay a black patch on random locations of the images to generate 3 variations of each subjects in addition to the original clean image. Fig. 3.1 shows a set of training images belonging to one subject. Another 1000 images are randomly picked and

Methods	Pair 1	Pair 2	Pair 3
KSVD	740	729	709
Proposed Method	**2**	**3.57**	**3.44**

Table 3.1: Average Ranking Using Proposed Method and KSVD on the Simulated Occlusion Training Dataset.

added to form the training set. I normalize each image to size 30×30.

For each subject X_i from the 187 subjects, the four training images $(X_{i1}, X_{i2}, X_{i3}, X_{i4})$ can form 6 possible pairs. I only assume 2 pairs $< X_{i1}, X_{i2} >$ and $< X_{i3}, X_{i4} >$ are labeled in the training stage, and the other 4 pairs are not used as constraints (hence simulating a more realistic situation where only partial labeling information is available). After I trained my dictionary on this dataset, I compute the codewords using (3.4) but now D is fixed. I define the similarity score between signal x_i and x_j as such:

$$S_{ij} = \|P^{ij} D(\alpha_i - \alpha_j)\|_F^2, \tag{3.31}$$

For the 6 pairs of each subject,I pick 3 pairs: $< X_{i1}, X_{i2} >$ (Pair 1), $< X_{i1}, X_{i3} >$ (Pair 2) and $< X_{i1}, X_{i4} >$ (Pair 3) to evaluate my proposed method against the KSVD baseline. I rank the similarity score of each of these three pairs among scores from all possible pairs formed by x_i and x_j. Table 3.1 shows the average ranking. I also show the histograms of the rankings in Fig. 3.2.First row results from the proposed method for Pair 1, Pair 2 and Pair 3 and second row:KSVD results for Pair 1, Pair 2 and Pair 3.

From these results, I can see that for Pair 1 (which is given an associated/labeled pair in the training stage, as mentioned above), I improve the rankings significantly compared with KSVD. Neither Pair 2 nor Pair 3 were labeled in the training stage, yet I are still able to generate much better ranking results compared to KSVD. Fig. 3.3 shows some atoms

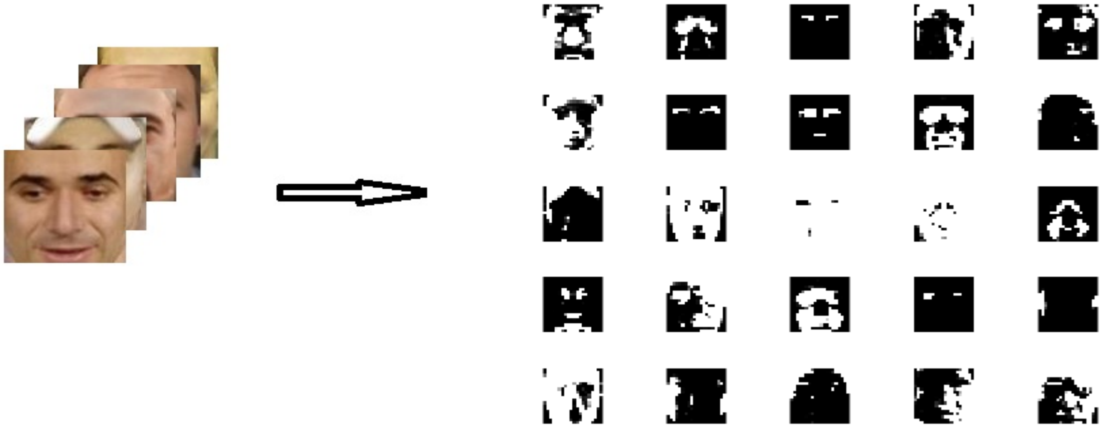


Figure 3.3: Left: Sample Cropped and Frontalized Face Images from LFW. Right: Sample Atoms of Learned D.

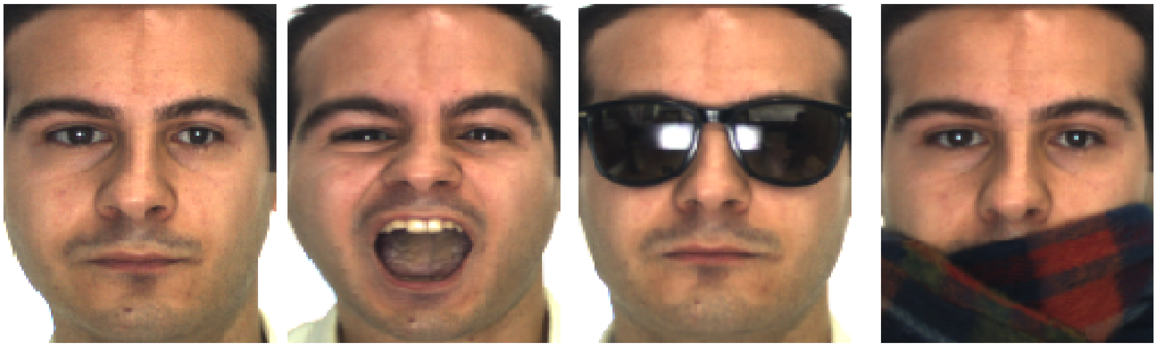


Figure 3.4: Sample images used in Experiment 2.

from the learned dictionary. As desired, each atom appears to represent certain part of the original face image.

Experiment 2: Face Retrieval Using AR Dataset

In the second experiment, I use real challenging images from the AR dataset. In the dataset, for each subject, there are 26 variations of different facial expressions, lighting conditions and occluded images. I exclude those images with lighting variations, resulting in 12 im-

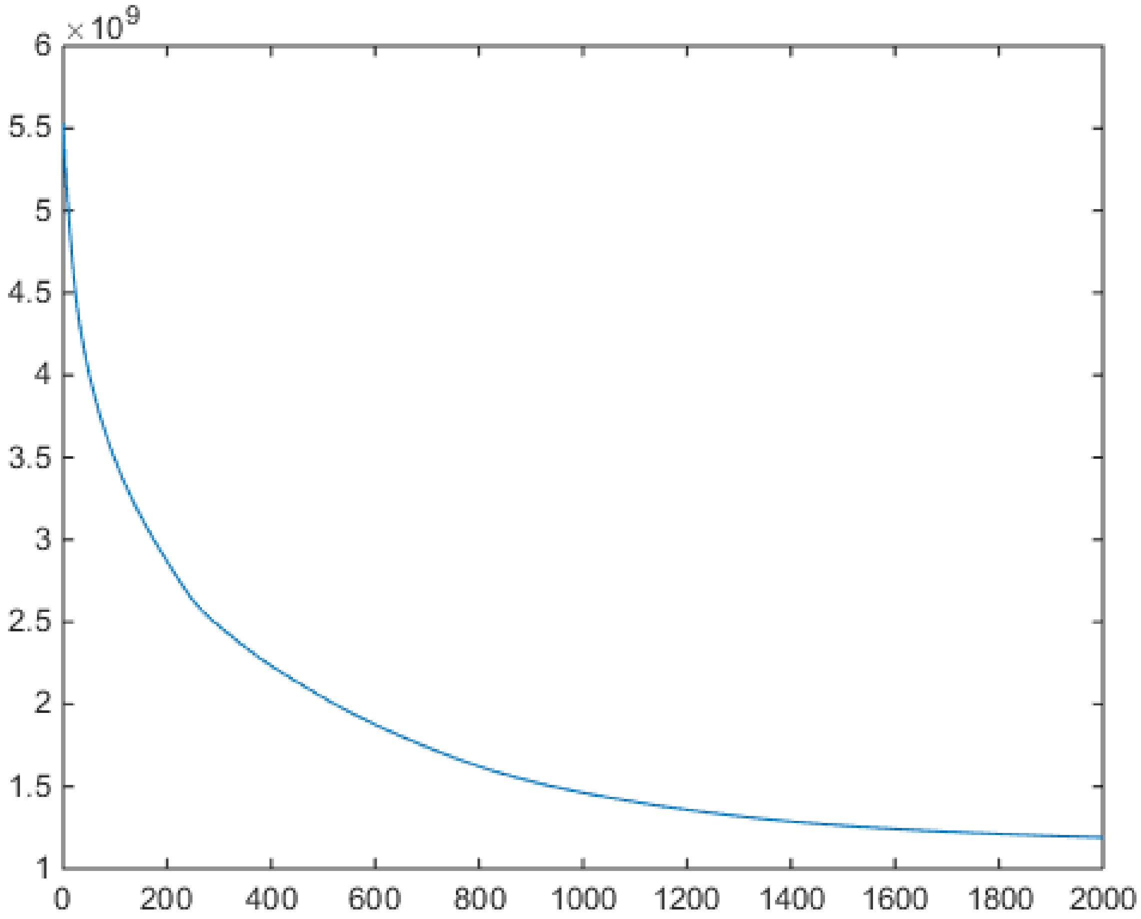


Figure 3.5: The Convergence Curve on the Simulated Occlusion Dataset Used in Experiment 1.

ages for each subject. Sample images are shown in Fig. 3.4. I form two disjoint sets containing 840 and 360 images respectively for training and testing. Then half of the images were randomly picked to serve as queries to evaluate face retrieval result using P@k defined as (3.32) on both the training set and the disjoint test set. I compare my method with both KSVD and DDLPC1 in Guo *et al.* (2013). My proposed method improved the $P@K$ measure significantly on both training and testing datasets, as summarized in Table

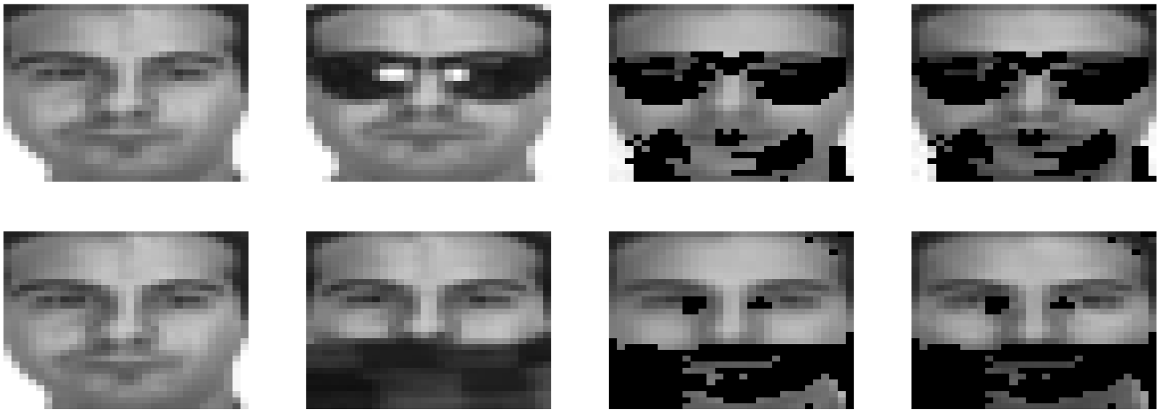


Figure 3.6: How P Capture the Occluded Part.

Methods	P@1 (training/testing)	P@3 (training/testing)	P@5 (training/testing)
KSVD	82.86%/82.78%	63.86%/65.93%	52.76%/51.89%
DDLPC1	84.44%/81.11%	68.15%/65.56%	56.77%/50.89%
Proposed Method	**100%/91.11%**	**77.06%/76.11%**	**61.43%/65.78%**

Table 3.2: Retrieval results on AR dataset.

3.2.

$$P@K = \frac{\#relevant\ images\ in\ top\ K\ retrieved\ results}{K} \tag{3.32}$$

Since P plays an important role in the model, I demonstrates how P impacts on the retrieval result. Table 3.3 shows $P@K$ values on the testing set using different φ. In general smaller value of φ tolerates more differences between two images. For this particular AR test set, the peak performance happens at $\varphi = 60\%$ to 70%.

φ	P@1	P@3	P @5
90%	89.44%	72.41%	59.78%
80%	91.11%	75.40%	63.44%
70%	**91.11%**	**76.11%**	**65.78%**
60%	**91.11%**	**76.11%**	**65.78%**
50%	91.11%	75.89%	64.46%
40%	91.11%	75.89%	68.89%

Table 3.3: $P@K$ Using Different Value of $varphi$.

Experiment 3: Face Verification Using AR and LFW Datasets

In this experiment, I evaluate my proposed method in face verification task. The definition of the task is defined as follows: given a pair of face images, decide whether they belong to the same person or not. I use the most common 'image-restricted' setting here: I only know whether the pair belongs to the same person, but the identity of the person is not given.

The LFW dataset provides a division of 10 folders with disjoint subject identities for cross validation the face verification result. I combine these 10 folders with the AR subset I used in experiment 2 3.1.4. Every face image is first cropped and normalized to size 30 by 30. In my experiment, for each independent evaluation,I randomly pick 300 matched pairs and 300 unmatched pairs from one LFW folder and the AR subset to form the test sets. Another 750 matched pairs and 750 unmatched pairs are randomly picked from the remaining images to serve as the training dataset to get dictionary D. The evaluation is repeated for 10 times. I compare my proposed method with KSVD Aharon *et al.* (2006) and DDLPC1 in Guo *et al.* (2013). Fig.3.7shows the ROC curve of my proposed method, the KSVD and DDLPC1, averaged on 10 evaluations.

The result shows that the face verification accuracy of my approach outperforms other

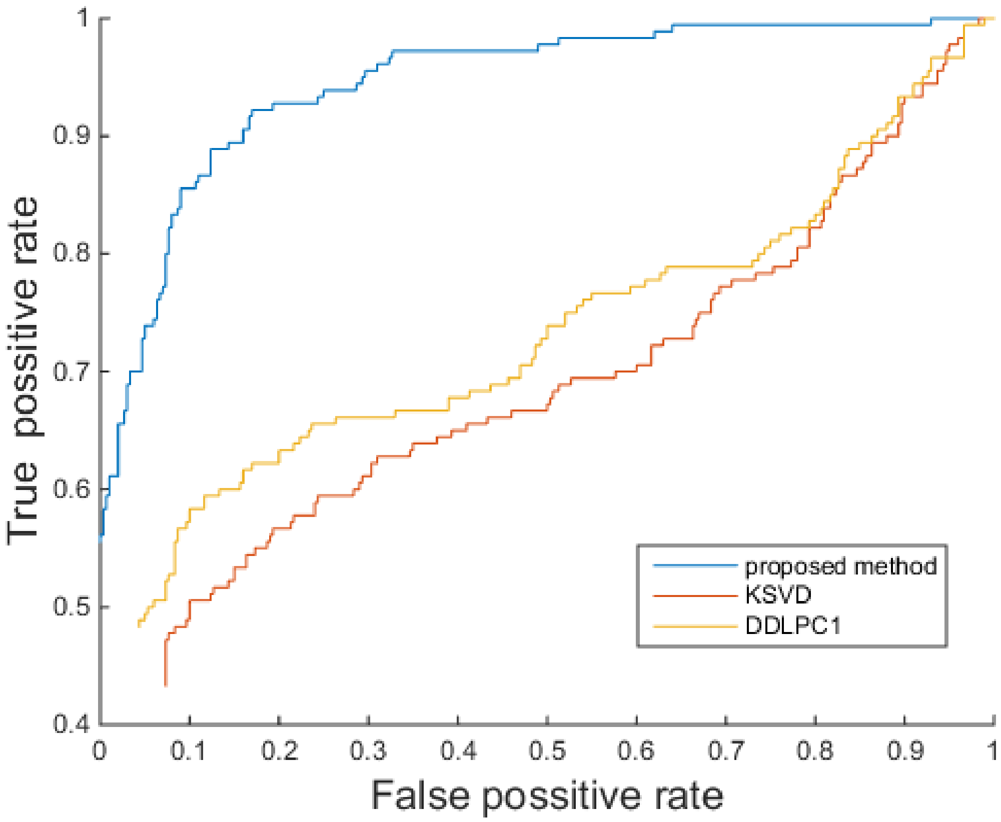


Figure 3.7: The ROC Curve of the Face Verification Experiment.

compared methods in this challenging experiment setting. Since the AR dataset I use here is very challenge because of the large areas of face occlusion, the performance of both KSVD and DDLPC1 decreases significantly comparing to what reported in Guo *et al.* (2013). It is also worth nothing that in the training stage, I use much less labels than the DDLPC1, because I only use matched pair labels, unmatched pairs are not explicitly labeled.

It is worth noting that although I do not explicitly encode the unmatch pair information into the objective function, but the result shows that the difference between faces from different person is actually enlarged compared to the KSVD dictionary representation. The average similarity score of unmatched pairs is 33.6% greater than that of pairs in the training set using my method. But this ratio for KSVD is only 10.1%. This ratio to some level indicates how well the sparse codewords can discriminate whether two samples image belongs to the same subject. The larger this ratio is, the better discriminative performance can be achieved.

3.1.5 Discussion

The theoretical analysis of convergency of the proposed algorithm has also been verified empirically: my proposed algorithm converges within the first 2000 iterations in both experiments. As an example, Fig. 3.5 illustrates the objective function of Eq. (3.4) monotonically decreasing during training for the simulated occlusion image dataset (experiment 1).

In terms of complexity, for a single iteration, the complexity of updating D is $O(MKN+hMK)$, the complexity of calculating P is $O(hM \log M)$, the complexity of updating A is $O(MKN + hM^2K)$, where h is the number of pairs from the same class, and $K \log K$ comes from the term $P^{ij}(\alpha_i - \alpha_j)$, which is a sorting problem. Hence, the overall complexity is $O(MKN + hM^2K)$.

Experiments show that my algorithm takes about 2000 iterations to converge, since the dictionary can be trained offline, my approach is still considered to be more valuable than the KSVD.

A percentage for quantifying the P matrix should be defined for the model. If the percentage is too large, the objective function of Eq. 3.4 tends to consider the whole $\|D(\alpha_i - \alpha_j)\|$ thus local distortion or occlusion will influence the result. If the percentage is too small, the learned D will lose discriminative power. I would like to point out that, given the nature of matching images, if e.g., 80% is used a good match is found, the e.g. 75% will also support a good (and in general better) match. Hence the key idea is on enabling the exclusion of some parts for matching while learning the dictionary. In my experiments, I used a fixed P ($\varphi = 70\%$) for both experiments only to illustrate this point. In practice, if computation cost is not a constraint, a search for the best percentage could also be incorporated into the algorithm. Fig. 3.6 illustrates how the automatically-figured-out P can capture the occluded part of a face, in a sample-adaptive fashion.

Although there are many other state-of-the-art methods for tasks like face retrieval and face recognition, I only pick KSVD and DDLPC1 to compared with for the reason that they are representative dictionary based methods. KSVD is a classical but powerful plain dictionary learning model and the DDLPC1 is state-of-the-art constrained dictionary learning model. While the main contribution in this chapter is to propose a new dictionary learning method, it is not fair to compare my method with other non-dictionary based method. As shown in Fig.3.3, my learned dictionaries preserve the part based property. And numerical experiment results show that associate information is successfully encoded in the learning of D.

3.1.6 Conclusion

I formulated a new dictionary-learning problem with limited number of pairwise association constraints and under the assumption that the association/similarity is defined only after some portions of the data samples are ignored. This would be a proper setting for applications like face matching/retrieval under partial occlusion or expression variations. An algorithm was designed and its convergence proven. Experiments demonstrate that, in comparison with relevant baselines, the proposed approach has clear performance gains. Future work along this direction includes seeking a more general P matrix that may support some transformation group.

3.2 Improving Robustness Of Random Forest Under Label Noise

3.2.1 Introduction

Random forest has a successful history in solving machine learning tasks. It has many appealing properties like simple structure, good ability to handle high dimensional data and relatively good speed performance in both training and testing. Recently, random forest has been used in many computer vision applications Bosch *et al.* (2007); Criminisi and Shotton (2013); Ren *et al.* (2015); Montillo *et al.* (2013); Richmond *et al.* (2015); Shotton *et al.* (2013) and generated many state-of-the-art results.

While random forest has many superior properties, it also has certain inefficiencies. Since in random forest the evaluation of each split node is based on the purity of the node, label noise in the training data may impact purity evaluation and hence degrading the performance. There are some existing efforts on improving the performance of random forest under noisy labels. A recent study Ghosh *et al.* (2017) showed that under symmetric label noise and large sample size at each node, the decision tree and random forest algorithms are robust; while in asymmetric noise, they are not robust in general. Unfortunately, in most real applications, label noise can be very asymmetric, and thus it remains an unsolved problem to improve robustness of random forest in face of label noise.

Label noise exists in many computer vision tasks. As the sizes of image datasets explode Krizhevsky and Hinton (2009); Deng *et al.* (2009), the manual work of labeling the images increases significantly. Nowadays the labeling of many datasets involves anonymous online users, thus the labeling precise is much more difficult to guarantee. While there may be many kinds of noise in real applications, in this work I deal with only the class-conditional random label noise, which assumes that the label noise depends only on the classes but not on the samples. Many types of real world label noise can be approximated by this simplified noise model.

In this work, I propose an approach to improving the performance of random forest under class-conditional random label noise. When random forest minimizes the classifier loss implicitly via recursively reducing the uncertainty of given training samples, there is no control over an overall classifier loss and its proper minimization. In Ren *et al.* (2015), a framework was proposed to incorporate a global loss function into the training of random forest. I adopt a similar strategy in my work and propose an estimator for the multi-class classifier loss under the assumed noise model. My approach not only preserves the structure and appealing properties of the classic random forest, but also makes it more robust to label noise. I evaluate my approach on 5 different datasets for classification, reporting promising results. The main contribution of this chapter is summarized as the following:

- To my best knowledge, this is the first attempt to consider asymmetric label noise in training random forest.
- I proposed a general multi-class noise-tolerant loss function that can be used in many noise settings.
- Extensive experiments on different datasets demonstrate the effectiveness of the proposed model under various noise settings.

In Section 3.2.2, I briefly revisit related prior work. In Sections 3.2.3, I present my proposed model and report experimental results. Further discussion summarizing important observations is given in Section 3.2.5. I conclude in Section 3.2.6.

3.2.2 Related works

Random Forest: Applications and Variations

Random forest Breiman (2001) is a widely-used machine learning model, which is an ensemble of a set of decision trees. In training, each decision tree is trained independently and outputs its own prediction. The final prediction of the forest is an average of each tree's outputs. The "randomness" mainly comes from two processes: the random selection of a subset from the training data for each tree and the random selection of a subset of features from the feature space.

Besides having a simple structure for implementation, random forest has many other good properties including being able to handle high-dimensional data, and fast training and testing. The appealing properties of random forest has made it one popular machine learning algorithm especially for computer vision applications Díaz-Uriarte and De Andres (2006); Lindner *et al.* (2015); Del Río *et al.* (2014); Belgiu and Drăguţ (2016). Work in Criminisi and Shotton (2013) demonstrate the success of random forest in computer vision domain especially in medical image analysis.In Ren *et al.* (2014),the author propose a better learning based approach using random forest. This approach regularizes learning with a locality principle based on two insights: for locating a certain landmark at a stage, 1) the most discriminative texture information lies in a local region around the estimated landmark from the previous stage; 2) the shape context and local texture of this landmark provide sufficient information. The author used a classic regression random forest to learn local feature mapping function for fast face alignment. Both works show promising results of using random forest in computer vision tasks.

In recent years, efforts have been made in order to further improve the classic random forest, resulting in many variations Menze *et al.* (2011); Montillo *et al.* (2011); Qiu and Sapiro (2015); Kontschieder *et al.* (2012). The following two recent efforts are most related to my proposed work. In Schulter *et al.* (2013),the author introduces a novel classification method termed Alternating Decision Forests (ADFs) , which formulates the training of Random Forests as a global loss minimization problem. The losses are minimized via keeping an adaptive weight distribution during the training over the training samples, which is similar to Boosting methods.During the training, each decision tree grows simultaneously in a breadth-first manner so the global loss can be measured at each state of the entire model. The author derived the new classifier and give evaluations on standard machine learning data sets. Furthermore, the author shows how ADFs can be easily integrated into an object detection application. A later work Ren *et al.* (2015) also uses the global loss to guide the learning process of random forest, but instead of measuring it at each tree growing stage, the loss is only used to modified the leaf nodes once the pre-trained random forest is given. The author in this paper claims that the learning and prediction of random forest is inconsistent: the learning of individual trees is independent but the prediction averages all trees outputs. As a consequence, he loss functions implied from these two processes are actually different which limits the fitting power of random forest. To alleviate such inconsistency the method proposed in this paper discard the old values stored in all tree leaves of a pre-trained random forest and relearn them through a global refinement.Both of these two works explored the possibility of adding global loss function into random forest.

Another state-of-the-art work is Kontschieder *et al.* (2015) in which the author proposed a novel structure which combines deep neural network and random forest structure. Different from using impurity function to split nodes in classic random forest, in the structure proposed, each node contains a routing probability driving by the output of a neural network. The structure is fully differentiable thus the forest can be updated using back propagation.

The afore-mentioned improvements do not consider label noise in training. Although by injection of randomness, random forest is more robust to outliers than most linear classifiers, it still suffers dramatic performance degradation especially under asymmetric label noise, as I will illustrate in Section 3.2.4.

Learning with Noisy Labels

In most real world applications, one needs to take care of label noise. Explicitly correcting label error for large datasets is expensive and thus may not be done with guarantee. Ergo it is desirable to improve the performance of machine learning under label noise. Recent years have seen intensified study on this regard. In Xiao *et al.* (2015), a framework was introduced to train Convolutional Neural Networks (CNNs) with only a limited number of clean labels and millions of noisy labels. The relationships among images,class labels and label noises are modeled with a probabilistic graphical model and integrate into an end-to-end deep learning system. In Jindal *et al.* (2017), a drop-out regularization was presented to deal with label noise in deep networks. These noisy label learning techniques are heavily rely on either huge deep networks and/or large amount of training data.

Among many attempts to deal with label noise, the work in Natarajan *et al.* (2013) provides a lightweight, general formula to generate an unbiased estimator of any bounded binary loss function under class-conditional random label noise, which also provides proofs of guarantee for risk minimization without any assumption of on the true noise distribution. This work appears to be by far the most general formulation of the problem of learning with noisy labels.

On the other hand, how to improve random forest under noisy labels is less studied. A recent work Ghosh *et al.* (2017) analyzed the robustness of decision tree based algorithms under label noise, where it was showed that gini index criteria based decision tree and random forest learning is robust under symmetric label noise and large sample size for binary classification. However, neither asymmetric label noise nor multi-class cases are studied.

Inspired by such efforts, in this work I attempt to address the problem of multi-class learning with noisy labels with random forest. I first introduce a global multiclass noise

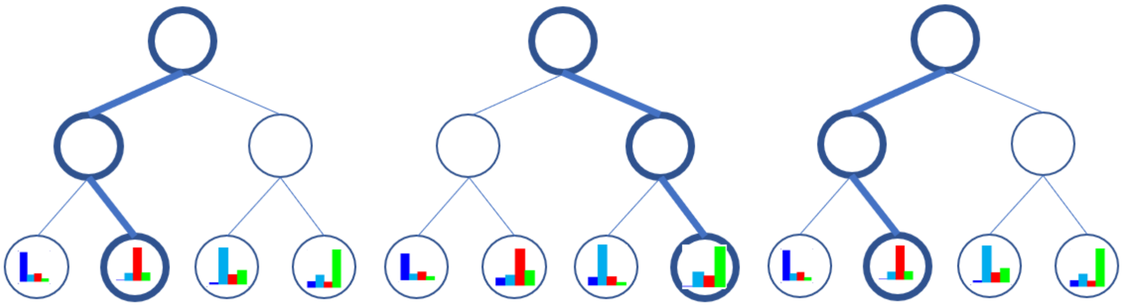


Figure 3.8: An Illustration of $S(X)$ and W from a Random Forest.

tolerant loss function and then incorporate it with the learning process of classic random forest. I show that the classic random forest suffers from asymmetric label noise and held extensive experiments to show that even use limited number of training samples with relatively high label noise rate, the proposed Noise Robust Random Forest still gives promising results on different datasets and noise settings.

3.2.3 The Proposed Noise Robust Random Forest (NRRF)

Basic Random Forest Formulation

To set the stage, I first briefly review the classic random forest approach Breiman (2001), which is an ensemble of K binary decision trees. In the training process, each tree is trained independently and generates a prediction. The prediction function of a single tree can be written as $\tau_t(x) : X \rightarrow R^N$, where X is the input feature space and $R^N = [0, 1]^N$ is the class distribution over the label space $Y = [1, ...N]$. In testing, each sample is sent to all decision trees, and the final predicted label $\hat{y}$ is an average of returned class distribution $p_n(y|X)$ from each tree:

$$\hat{y} = argmax \frac{1}{K} \sum_{N} p_n(y|X) \tag{3.33}$$

During training, a random subset of the training data is fed to each decision tree (i.e., bagging). For each single decision tree, the training data relies on randomly sampling a subset of the features and splitting the training samples at each node such that the training samples in the newly created child nodes is pure according to some impurity measurements. Each tree is grown until some stopping criterion, e.g., the maximum tree depth, is reached and the class probability distributions are estimated by the number of labels from each class in the leaf nodes. Commonly-used impurity measurements are entropy and gini index. Through out this chapter, I pick gini index as the impurity measurements for both classic random forest and my proposed NRRF. An N class gini index of a node D of a tree is defined as:

$$l_{gini}(D) = 1 - (p_1^2 + p_2^2 ... + p_N^2) \tag{3.34}$$

where

$$p_n = \frac{\{(x, c(x)) \in D : c(x) = c_n\}}{|D|} \tag{3.35}$$

Then the best splitting for a node is chosen by minimizing

$$I_{score} = \frac{|L|}{|L| + |R|} l_{gini}(L) + \frac{|R|}{|L| + |R|} l_{gini}(R) \tag{3.36}$$

where L and R are the left and right child node of D respectively.

Particularly in a two-class case, let $p = n_0/n, q = n_1/n$ be the fractions of the two classes at node v under noise free samples, where n_0, n_1 are the number of samples in class 0 and 1 respectively, and $p + q = 1$ holds for each node. The gini impurity then becomes $G_{Gini}(v) = 2pq$. Under symmetric label noise rate θ, gini impurity gain becomes:

$$\begin{aligned} G^{\theta}_{Gini}(v) &= 2p^{\theta}q^{\theta} \\ &= 2[((1 - 2\theta)p + \theta)((1 - 2\theta)q + \theta)] \\ &= 2pq(1 - 2\theta)^2 + (\theta - \theta^2) \\ &= G_{Gini}(v)(1 - 2\theta)^2 + (\theta - \theta^2) \end{aligned} \tag{3.37}$$

Similar expressions hold for $G^{\theta}_{Gini}(vl)$ and $G^{\theta}_{Gini}(vr)$: the left and right child node of v. The (large sample) value of criterion or impurity gain of f under label noise can be written as:

$$\begin{aligned} & gain^{\theta}_{Gini}(f) \\ =& G^{\theta}_{Gini}(v) - [\alpha G^{\theta}_{Gini}(vl) + (1-\alpha) G^{\theta}_{Gini}(vr)] \\ =& (1-2\theta)^2 [G_{Gini}(v) - \alpha G_{Gini}(vl) - (1-\alpha) G_{Gini}(vr)] \\ =& (1-2\theta)^2 gain_{Gini}(f) \end{aligned} \tag{3.38}$$

where α represents $\frac{|vl|}{|vl|+|vr|}$.Thus for any $\theta \neq 0.5$, if $gain_{Gini}(f1) > gain_{Gini}(f2)$, then $gain^{\theta}_{Gini}(f1) > gain^{\theta}_{Gini}(f2)$. Which means that a maximizer of impurity gain based on gini impurity under clean samples will also be a maximizer of gain under symmetric label noise, under large sample limit. The work in Ghosh *et al.* (2017) studied the performance of random forest under noise rate varied from 0% to 40% with different sample sizes, the conclusion is that while the above proof holds for symmetric label noise, it is hard to arrive at same conclusion for asymmetric noise. In general, decision tree based classifiers are not robust under asymmetric noise. Similar to (3.37), when the noise is asymmetric, under large sample limit, the gini impurity at a node v becomes:

$$\begin{aligned} \hat{G}_{Gini}(v) &= 2\hat{p}\hat{q} \\ &= 2[((1-\theta_0-\theta_1)p+\theta_1)((1-\theta_1-\theta_0)q+\theta_0)] \end{aligned} \tag{3.39}$$

where $\hat{p}$ and $\hat{q}$ are observed noisy fractions of class 0 and class 1. From (3.39) I can approximately recover the "clean" gini impurity by:

$$G_{Gini}(v) = \frac{2[\hat{p}-\theta_1][\hat{q}-\theta_0]}{1-\theta_0-\theta_1} \tag{3.40}$$

I named this as "modified gini index" because the class fractions in each node is modified to approximate the gini impurity for clean samples. It's worth noting that the above equation only holds under large sample limit. This sample limit is set as a parameter in the learning stage. As the splitting process goes, at some point the number of samples in a node will fall below the sample limit. After the large sample limit is reached, the above approximation is not guaranteed.

Based on above discussion, it is not easy to remove the label noise impact within the structure of classic random forest. It is intuitive to think about adding additional training steps to solve this problem. In the following sections, I explore a global loss function based method.

Introducing a Relaxed Multi-class Noise-tolerant Loss

Let $D = (X, Y)$ be the distribution of the sample space, where $X \in \mathbb{R}^d$ and $Y \in \{1, ..., N\}$, and N is the number of classes. For any sample drawn from D, it has the form (X_n, Y_n). If I inject random classification noise to the labels, I obtain $(X_n, \tilde{Y_n})$. The classification noise can be defined as follows:

$$\rho_{ij} = P(\tilde{Y} = j \mid Y = i) \quad \forall i, j \in \{1, ..., N\} \tag{3.41}$$

I assume that the noise rates are known during the training stage, although it is not necessary in practice. In Natarajan *et al.* (2013), a method of unbiased estimation for the loss was proposed, which is stated as follows:

Lemma 3 (Method of Unbiased Estimators Natarajan *et al.* (2013))**.** *For a 2-class classification problem ($y \in \{-1, 1\}$), let $l(x, y)$ be any bounded loss function. Define $\tilde{l}(x, y)$ as follows:*

$$\tilde{l}(x, y) = \frac{(1 - \rho_{(-y,y)})l(x, y) - \rho_{(y,-y)}l(x, -y)}{1 - \rho_{(-y,y)} - \rho_{(y,-y)}} \tag{3.42}$$

Then $\tilde{l}$ is an unbiased estimator for l.

This method is surrogate-loss based, which exploits a symmetry condition on the loss function such that it can provide an unbiased estimator of the non-noisy risk. Later researchers extended this idea into the multiclass scenario Patrini *et al.* (2016), as given below:

Lemma 4 (Backward Corrected Loss Patrini *et al.* (2016))**.** *Assume that the noise rate matrix* P *is non-singular. For a given loss function* l*, the backward corrected loss* $\tilde{l}$ *can be defined as:*

$$\tilde{l}(x,y) = \mathrm{P}^{-1}[l(x,1), l(x,2), ..., l(x,N)]^T \tag{3.43}$$

where

$$\mathrm{P} = \begin{bmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{N1} & \rho_{N2} & \cdots & \rho_{N1} \end{bmatrix} \tag{3.44}$$

Then $\tilde{l}$ *is an unbiased estimator for* l*.*

The loss function in Eq. (3.43) is a multiclass generalization to the one in Eq. (3.42). When $N = 2$, they are equivalent.

Inspired by Natarajan *et al.* (2013), I propose a simpler loss function for training with noisy labels:

Lemma 5. *Let* $l(x,y)$ *be the loss function for noisy-free dataset. The loss function for training with noisy labels can be defined as:*

$$\tilde{l}(x,y) = \frac{(1 - \sum_{i \neq y} \rho_{iy}) l(x,y) - \sum_{j \neq y} \rho_{yj} l(x,j)}{1 - \sum_{i \neq y} \rho_{iy} - \sum_{j \neq y} \rho_{yj}} \tag{3.45}$$

Then $\tilde{l}$ *is an estimator for* l*.*

Although Eq. (3.43) is a natural extension of Eq. (3.42), there is no guarantee that the matrix P is non-singular. Comparing to Eq. (3.43), my loss function in Eq. (3.45) does not have such constraint thus it can be viewed as a relaxed version of (3.43).

I name Eq. (3.45) ”relaxed multi-class noise-tolerant loss”. It can be minimized using SGD Zhang (2004).

Formulation of Noise-Robust Random Forest

I now formulate the proposed Noise-Robust Random Forest. Similar to Ren *et al.* (2015), first I define a selecting function $S(x)$: the routing process of a sample in a forest can be viewed as mapping the input x onto selected leafs. $S(x)$ is binary and has the same dimension of number of leaves. Each dimension indicates whether the input finally arrives at the leaf node or not (1 or 0). Figure 3.8 illustrate the structure of a random forest with 3 trees and corresponding $W, S(X)$ for a particular input.

Each leaf node of the RF represents a prediction (e.g., a estimated distribution for classification or continuous values for regression). I denote each leaf as a vector w_i, and all the leaf nodes in a random forest can be written as $W = [w_1, w_2, ...w_T]$ where T is the number of leaves.

Having $S(x)$ and W, the prediction of the random forest defined in Eq .(3.33) can be rewritten as :

$$\hat{p} = WS(x) \tag{3.46}$$

With Eq. (3.46) and Eq. (3.45), I formulate the objective function of NRRF in the following form:

$$\begin{aligned} &\min \frac{1}{2}||W||_F^2 + \lambda \sum_{i=1}^{M} \tilde{l}(\hat{y}_i, y), \\ s.t. \quad & y_i = WS(x), \forall i \in \{1, ..., M\} \end{aligned} \tag{3.47}$$

where M is the number of training samples, λ is a control parameter trading off between the L_2 norm of W and the noisy loss.

Throughout my experiments, I use multiclass log loss as $l(t, y)$, which is defined as:

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} log(\hat{p}_{i,j}) \tag{3.48}$$

where M is the number of observations, N is the number of classes, log is the natural logarithm, $y_{i,j}$ is 1 if observation i is in class j and 0 otherwise, and $\hat{p}_{i,j}$ is the predicted probability that observation i is in class j. There are other loss functions like hinge loss and tanh loss. Although I do not explicitly explore all of them in this chapter, I believe in principle the performance of NRRF should be similar under such losses.

The complete NRRF approach is shown in Algorithm 1. I initialize the forests using classic RF and update W iteratively using Eq. (3.45) until convergence. For testing, a sample will first go through the classic RF to calculate its corresponding $S(X)$. The final prediction $\hat{y}$ will be generated using Eq. (3.46).

Algorithm 1: Learning the NRRF

```
Input      : Training set X, Training labels Y
Parameters: Parameters for RF(Max depth,Min samples etc), learning rate $\lambda$
Output     : Trained NRRF
while Depth < Max depth and # of samples > Min samples do
    Splitting current node using gini index
end
while not converge do
    Calculate loss using eq(3.45);
    Compute gradient of the loss: $grad(loss)$;
    Update leaf nodes $W$ : $W- = grad(loss) * \lambda$
end
```

3.2.4 Experiments

I evaluate my proposed NRRF on classification tasks on four standard machine learning dataset and Cifar10. I compare my proposed NRRF model with the classic random forest model (referred as RF in the rest of this section) and the Global Refined Random Forest Ren *et al.* (2015) (referred as GRRF in the rest of this section). I study the impacts of different label noise distributions on all three models.

Experiment settings

In this section I describe how I set up each experiment. First I come up a way of generating ”noisy” training set. Since all the standard dataset are noise-free in labels, I artificially introduce random label noise into the training splits. In my implementation, I define ρ_i to be the total probability of a true class i sample to be observed as class other than i. For implementation simplicity, I assume that the probability of a sample in class i to flip into any other class is equal. For example, in a 10-class classification task, a $\rho_1 = 0.36$ means the probability of class 1 samples get flipped into another class is $\frac{\rho_1}{9} = 0.04$, so given ρ_i, the ith row of matrix P can be calculated accordingly.

I assume the noise vectors are given in all my experiments. I want to make it clear that it is not necessary in practical. There are multiple ways to estimate noise levels, two most recent methods are described in Natarajan *et al.* (2013) and Patrini *et al.* (2016) respectively. The first method uses cross validation information and the second one estimate each component of matrix P just based on noisy class probability estimates from the output of a softmax layer.

I set the rest of the experiment parameters as the following. Unless stated otherwise, the number of trees is set to 50. The number of random features sampled and tested in each node is set to the square root of the feature dimension, as recommended in Breiman (2001). The minimum sample number for split is set to 10. The maximum depth of the tree is set to 10,15 or 25, depending on the size of the training data. I use the given training/testing for most of the dataset. For datasets without standard training/testing split, I randomly split the dataset into 60% for training and 40% for testing.

Most existing noise classification makes the assumption that the noise is content based, using information like a cat is more likely to be mislabeled as dog than car. But in general,

Dataset	MNIST	Cifar10	letter	covtype	usps
# Train Samples	60000	50000	15000	348607	7291
# Test Samples	10000	10000	5000	232405	2007
# classes	10	10	26	7	10
# Feature dimensions	784	512	16	54	256

Table 3.4: The Properties of Datasets Used in the Experiments.

this kind of information is not always available. Since I try to approach the problem in a different way: using only noise rate information, thus it is different from most existing methods. The key contribution I intended to make is to explore a new way of improving RF with label noise information. Thus I compare my proposed NRRF with two other methods: RF and GRRF.

Properties of Datasets Used

I use 5 widely used dataset to evaluate my proposed NRRF: MNISTLeCun *et al.* (1998), Cifar10 Krizhevsky and Hinton (2009), letterFrey and Slate (1991), covtypeBlackard (1998) and uspsHull (1994).The properties of these datasets can be found in the corresponding cited papers. In experiment I and IV, I evaluate my proposed NRRF and other comparing models on all five datasets. In experiment II, III and IV, I use the well known MNIST dataset.

Experiment I: Comparison of RF, GRRF and the Proposed NRRF

In this experiment, for each dataset I generate 10 sets of class conditional random noises and flip the samples in the training dataset accordingly. For all of RF, GRRF and proposed NRRF, I use the training parameters described in experiment settings. The learning rate is

Method	RF	GRRF	NRRF
MNIST	0.1986 ± 0.05	0.1845 ± 0.05	0.1142 ± 0.02
letter	0.2038 ± 0.05	0.1977 ± 0.05	0.1570 ± 0.05
covtype	0.2306 ± 0.04	0.2276 ± 0.03	0.1879 ± 0.01
usps	0.1862 ± 0.08	0.1759 ± 0.08	0.0804 ± 0.01
Cifar10	0.5416 ± 0.06	0.5322 ± 0.04	0.5013 ± 0.03

Table 3.5: Results on Classification Task.

set to 0.001 for updating W in NRRF. I report both the average classification error rate and the testing loss with standard deviations resulting from 10 repetitions of each noise setting. Table 3.5 shows the classification error rate on 5 datasets.

Table 3.6 shows the average training and testing loss using multiclass log loss. Interestingly, my proposed NRRF have a higher training loss though out the experiments. This is expected because by adding the global noise tolerant loss function I are explicitly telling the NRRF model that there are noise in the training set although I only know the information about the noise rate not exactly the particular noisy samples. As a result, the NRRF tries to figure out ”how much” it should be ”fitted” in the training, thus the training loss goes up because some of the training samples are treated as noise and are ”less fitted”.

The standard deviations of classification error is quite small among all the experiments for NRRF, which means that the NRRF is quite robust under different noise setting. Instead, the classic RF and the comparing GRRF have higher std. In the next experiment I will analyze the performance of RF under different class conditional noise.

Method	RF	GRRF	NRRF
MNIST	0.681/0.736	0.428/0.557	0.616/0.425
letter	0.644/0.994	0.418/0.862	0.957/0.675
covtype	1.030/0.087	0.840/0.079	1.034/0.073
usps	0.292/0.375	0.451/0.530	0.312/0.309
Cifar10	0.925/1.764	0.913/1.527	1.106/1.686

Table 3.6: Training/Testing Loss.

Experiment II: Impact of Different Label Noise Distribution on RF and NRRF

As proofed in Jindal *et al.* (2017), RF is robust to symmetric label noise under large sample size limit. I observed in my experiment I that when the noise rate in each class are close to uniform distribution, RF and proposed NRRF intend to perform similarly which verifies the proof in Jindal *et al.* (2017). Here I take a closer look at the performance of RF under asymmetric label noise distributions.

In this experiment, I only use MNIST dataset. I generate 3 groups of noise rate vectors. According to their values I give them different names:

1. ”Uniform”:Noise rate in each class is equal to 0.2
2. ”One peak”:One class has significantly higher noise rate then other classes, the peak noise rate is 0.8.
3. ”Two peaks”:Two classes has significantly higher noise rate then other classes, the peak noise rates are 0.8.

I randomly generate 10 noise rate vectors that satisfied the conditions for each group. The performance is reported in terms of the average classification accuracy in each group.

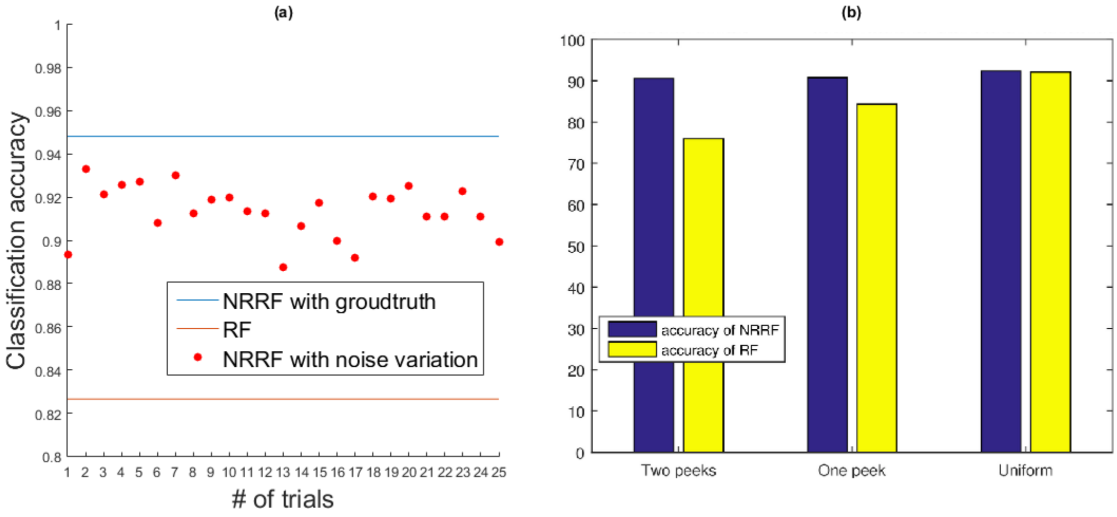


Figure 3.9: (a) The Performance of NRRF under Variations of True Noise Rate; (b) The Performance Differences on Different Label Noise Distributions

For ”One peak” group, the average performance gain of NRRF over RF is 12%; for ”Two peaks” group, the average performance gain is 9%; for ”Uniform” group, the performance gain is 2.5%. Clearly, I can see that with one peak and two peak noise, the performance of RF degraded significantly comparing to the uniform group. Figure 3.9(b) plots the mean accuracy bar charts under 3 groups of noise rate vectors.

One the other hand, my proposed NRRF have a stable performance around 92% test accuracy in all three groups, which is a appealing property in real world applications.

Experiment III: Learning with Uncertain Noise Rate

I have assumed given noise rate in above discussion. In practice, I can at best know an estimate of the true noise rate for each class. Hence, I now evaluate how the proposed NRRF perform without knowing precise noise rate (but using only a close estimate). I start by picking an asymmetric label noise vector to serve as the ground truth and generating the noisy training set. Then I add some small random noise between $[-0.1, 0.1]$ to the ground

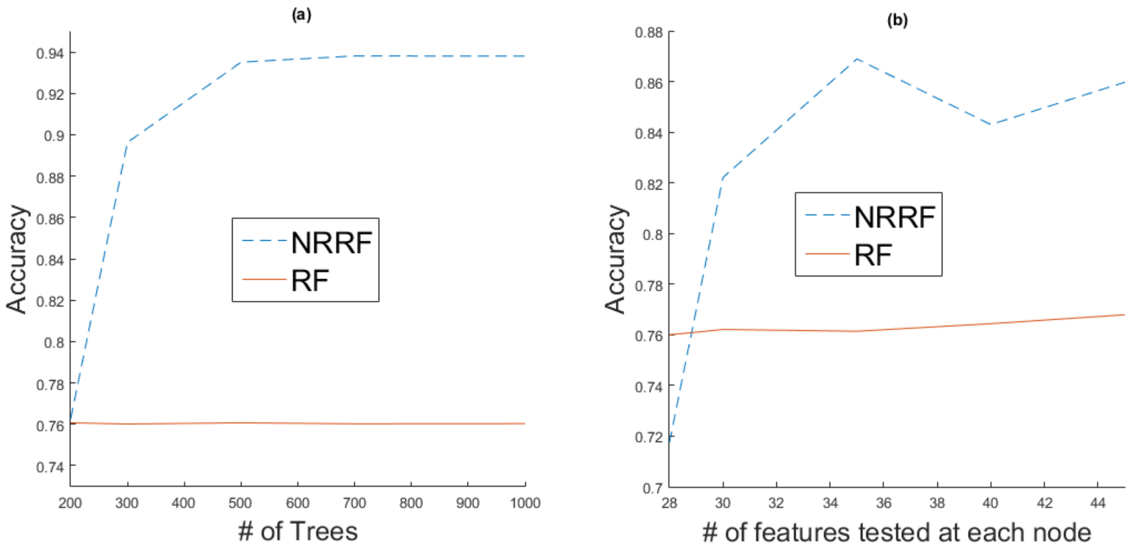


Figure 3.10: The figures show the performance and accuracy of NRRF: (a) the accuracy curve by varying number of trees; (b) the accuracy curve by varying number of features tested at each node.

truth noise vector:

$$\rho_{true} = [0.5, 0.3, 0.2, 0, 0, 0, 0, 0.6, 0, 0]$$

By "adding noise" for different trials (I do 25 trails here), I obtain different variations of the ground truth noise vector.

Then I feed the above noise rate vectors to NRRF, Figure 3.9(a) shows the performance (in terms of classification accuracy) of NRRF. The blue line is the NRRF performance when given ground truth noise rate vector; the red line is the performance of RF. The results of the trials fall into the space between the two lines. Obviously, with various small variations of the ground truth noise rate, the NRRF is still able to generate better predictions compared to the classic RF. Hence I argue that my proposed NRRF is robust under small uncertainty of the underline ground truth noise rate. And obviously if large size of clean test data is available, one can use cross validation to find the best noise rate that generates the optimal

Dataset	MNIST	letter	covtype	usps	Cifar10
Loss A	0.1142	0.1570	0.1879	0.0804	0.5013
Loss B	0.1208	0.1578	0.2034	0.1031	0.5013

Table 3.7: Test Error Rate Using Different Loss Functions.

prediction results.

Experiment IV: NRRF with Different Loss Function

In this experiment I compare the performance of my relaxed multiclass noise tolerant loss function (referred as ”Loss A” in the rest of this section) with the one proposed in Ghosh *et al.* (2017) (referred as ”Loss B” in the rest of this section). Table 3.7 shows the error rate on 5 datasets using both loss functions when the noise matrix P defined in Eq.(3.44) is invertible. I can see that with my proposed Loss A and Loss B, the NRRF generates similar results in most datasets when the inverse of P exists. The limitation of Loss B is that P must be invertible. Now I show that when P does not have its inverse, my proposed Loss A can still generate improved results comparing to the classic RF when Loss B fails. Taking the MNIST dataset as an example, I generate 10 different P matrix which are singular or close to singular. My proposed loss A still gives an average of 10% improvement over RF while Loss B completely failed.

Analysis of RF Parameters

In this section I use the well known MNIST dataset and analyze the impact of two parameters related to RF: number of trees and number of features tested at a split node. The other parameters in this section unless otherwise noted ,are the same as described in experiment settings.

Figure 3.10(a) shows the classification accuracy curves of RF and my proposed NRRF by varying the number of trees. The performance of my proposed NRRF keep boosted as the number of trees increases, but the accuracy of classic RF stays and oscillates between 0.74 to 0.76.

Figure 3.10(b) shows the classification accuracy curve of RF and NRRF by varying the number of features tested at each split node. Both RF and NRRF have a performance gain as the number of tested features increases, but for NRRF the increase is more substantial.

3.2.5 Discussion

There are several important observations from my experiments. First, the my proposed loss function converges very fast in training. Usually I only need about 10 iterations for getting optimal results. I implement my proposed NRRF using python, the RF part is written using RandomForestClassifier from the sklearn package. For a random forest with 50 trees, 10 max depth, it has approximately $25k$ leaf nodes. Taking MNIST dataset as an example, the size of W will be about $10 \times 25k$, each iteration for updating w is less than 2 minutes. The total running time of the NRRF is the time of fitting an RF plus updating W. For MNIST experiment, this running time is about 17 minutes.

Second, as I stated in Section 3.2.3, my proposed global loss function Eq (3.45) is a relaxed version of Eq (3.43). I compared performance of NRRF under either version. The NRRF using my proposed loss function consistently gives an average of 2% accuracy gain over the loss function defined in Eq (3.43).

3.2.6 Conclusions and future work

In this work, I present a noise robust random forest model that improves the fitting power of classic random forest under class conditional random noise.This is the first work of learning noisy labels using RF structure.I evaluate my proposed NRRF on five datasets.

The results show that it outperforms the classic RF and Global refined RF under asymmetric class conditional random noise. The proposed NRRF is robust under different settings of noise rate and is very light weight in terms of implementations and running time. The NRRF perfectly preserved the structure and properties of RF meanwhile fit better to noisy data.

The relaxed multiclass noise tolerant loss function I proposed is verified to be a appropriate loss function by intensive experiments. I claim that this function is a relaxed version of Eq (3.43) and is more powerful in real application because of removing the constraint.

As in this chapter and prior related works, the improvements of RF are all done by adding a feedback. The feedback can be a loss function or something else. This process is actually adding another ”layer” or learning process to the classic random forest.

A more compact and intuitive way of improving the RF may be modifying the impurity function when learning it. There is a possibility that a noise tolerant impurity function can be derived then the learning process of the improved RF is exactly the same as learning a classic RF. As briefly discussed in section 3.2.3, simply modify the gini impurity measurements based on the noisy rate may not be a good idea, but other impurity measurements like cost sensitive measurement may be considered. This is an interesting direction to explore and may be my future work.

Chapter 4

SIMULTANEOUS CLUSTERING AND FEATURE SELECTION VIA LDA MODELS

In the era of Internet and social media data explosion, analyzing multi-modal data is an important task to gather information from rich content. Image tweets/microblog posts which have image and/or video embedded make the content more vivid. According to (Chen *et al.*, 2015), multimedia form attracts larger viewership than text only posts. On the other hand, many real-world applications involve high-dimensional data in various representations/views that provide related and complementary information. For example, one image can be described by its different views like color histogram, texture information, SIFT descriptor and so on. As the views are in general high-dimensional and may provide complementary information, selecting most relevant features from such data is often a necessary step for further analysis tasks. Based on above discussion, I aim to embed feature selection into LDA topic model. When multi-view LDA model can capture the correlations between each view and the text/annotations, feature selection step further make use of the correlation between views to build a more precise model of multi-modal data.The approach integrates unsupervised clustering and multi-view feature selection into a unifying formulation so that relatedness of the views, clustering of the data and importance of the features are all simultaneously considered.

In this chapter,I propose a new model that incorporates the feature selection procedure into the existing multi-view LDA algorithm framework to analyze social media data. To find a solution under the proposed model,I apply an iterative procedure to switch between feature selection and LDA. The evaluation of the approach is done using controlled dataset.

4.1 Introduction

In the era of Internet and social media, multi-view data source like twitter, instagram and flicker are seeing phenomenal growth. Given an image with multiple annotations, one way to described the correlations between the images and annotations is modeling them using topic model. The research problem is formally defined as follows: given a datasets of images and corresponding annotations, analyze hidden correlations between different views, via feature selection LDA model. The basis of our proposed approach is two-view lda model. While existing work focuses on purely topic modeling using different graph structures,I improve this modeling problem from the following angle: extract most important features from different views of data beyond simple concatenation of them.

The contributions of our work are summarized as follows: Firstly,I propose a new LDA model by adding feature selection component. Secondly, to solve the new formulation,I propose an efficient algorithm and evaluate via experiments its efficiency and effectiveness.

4.2 Related Work

In this section,I review briefly related research, and discuss the difference between the reviewed work and our proposed method.

4.2.1 Topic Modeling in Data Mining

Topic modeling is a convenient way of analyzing large amount of unclassified documents. The definition of a topic is a group of words that often appears together. A successful topic modeling will learn the connections of words with similar meanings and distinguish words with different meanings.

There are various kind of topic models, such as Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA), Correlated

Topic Model (CTM). These topic models have successfully improved classification accuracy in the area of discovering topic modeling. Among the above mentioned models, the most used and studied one is the latent Dirichlet allocation (LDA). LDA is a generative probabilistic model was first introduced by Blei *et al.* (2003) for single view data, to be specific, collections of discrete data such as text corpora. The original LDA is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a nite mixture over an underlying set of topics. Each topic is modeled as an innite mixture over an underlying set of word probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document.

Later, many work adapted the original LDA for multiview data. In Blei and Jordan (2003), the author extend the LDA to correspondence LDA model (corr-LDA), which can be used for modeling the relationship between two views: images and their tags/annotations. In Chen *et al.* (2015), the author claims that among multiple views that can describe an image, each particular tag may associate with only one view. This idea further extends LDA model into a three view graph model. Figure.4.1 shows the developing of LDA models from single view to multiview with (a): The original single view LDA model used for modeling textual data; (b) corr-LDA model used for annotated data; (c) Multi view LDA model, capture complex relationships between different views.

Although there are many variations of LDA models, the fundamental assumption used in these models are the same. The LDA models assume that the corpus are generated by a group of hyper parameters, a typical generative process of two-view data (visual view and text view) is described below:

1. Draw topic proportions $\theta \sim Dirichlet(\alpha)$.
2. For each visual word w_n, $n \in \{1, , N\}$:
    (a) Draw topic assignment $z_n \mid \theta \sim Multinomial(\theta)$.

(b) Draw visual word $w_n \mid z_n \sim Multinomial(\phi_{z_n})$

3. For each textual word $t_m, m \in \{1, , M\}$

(a) Draw discrete indexing variable $y_m \sim Uniform\,(1, ..., N)$

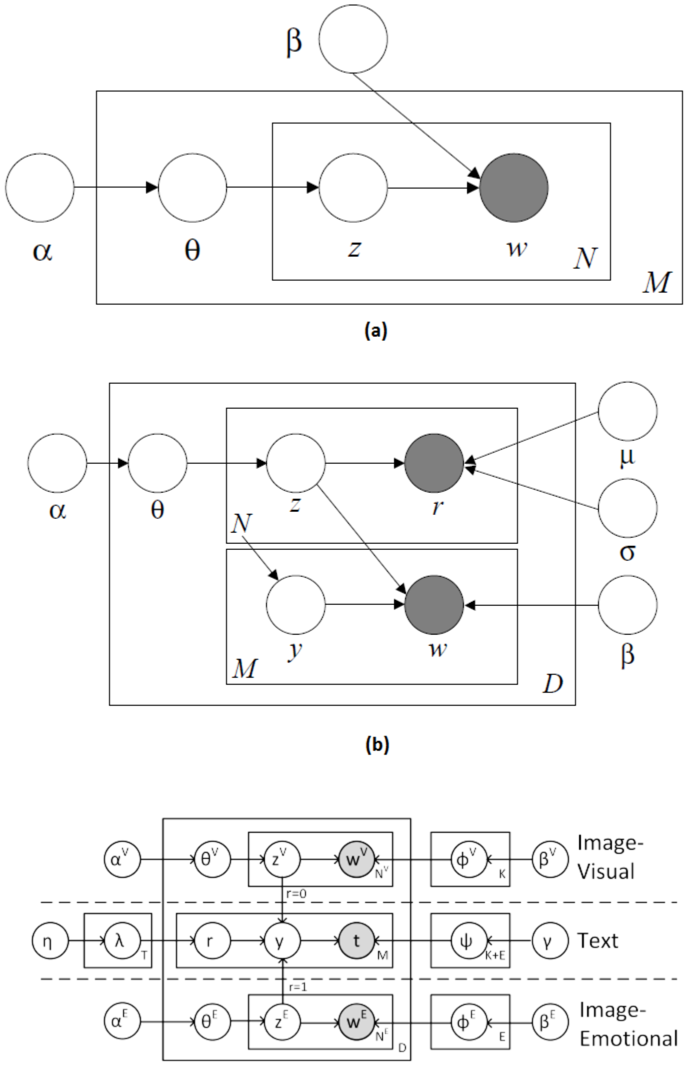
(b) Draw textual word $t_m \sim Multinomial(\psi_{z_{y_m}})$

Through out this chapter, I use this generative model to generate experimental data. This generative process is related to corr-LDA described in Blei and Jordan (2003), the difference is that, instead of assuming each textual word is associate with one region of the image,I assume that each textual word is associate with one visual feature of the image.I refer to this model as two-view LDA in the following sections.

4.2.2 Unsupervised Feature Selection

Feature selection, which reduces the dimensions of the data by selecting only a subset of most relevant features, has been proven in (Yu and Liu, 2003; Xing *et al.*, 2001) as effective for addressing the issue of curse of dimensionality, as written in Bellman (2015). Curse of dimensionality occurs in many machine learning problems involving high-dimensional data.

Existing feature selection methods may be broadly divided into two categories: Supervised feature selection (Guyon and Elisseeff, 2003; Song *et al.*, 2007; Sotoca and Pla, 2010) and unsupervised feature selection(Mitra *et al.*, 2002; Li *et al.*, 2012; Cai *et al.*, 2010). In many real-world applications, labeled data are often limited while unlabeled data are abundant (e.g., considering unlabelled on-line images), and thus, unsupervised feature selection has attracted attention. Unsupervised feature selection may select features that can better preserve some measures of data similarity in the feature space, as mentioned in research from He *et al.* (2005); Zhao and Liu (2007). In recent years, sparse learning has been employed for unsupervised feature selection. Sparse learning based approaches usually rely



(c)

Figure 4.1: LDA and Its Variations

on sparse coding enabled by critically labeled features using clustering strategy. Works include Unsupervised Discriminative Feature Selection (UDFS) in Yang *et al.* (2011b)and Multi-cluster feature selection (MCFS) in Cai *et al.* (2010). Recently, Embedded Unsupervised Feature Selection (EUFS) was introduced in Wang *et al.* (2015), incorporating both sparse learning and feature selection into a single step.

One common observation in the most of the existing work is that, while modeling the correlation between different views of a given document, few were discussed concerning the features used during the modeling processes. In particular, the number of dimension and the kinds of features used are all empirical. When several features are derived, all which are concatenated into one vector to be the final feature vector. For example, in Chen *et al.* (2015), saturation, brightness, hue, color names, pleasure, arousal, and dominance are features that are used, ending with the final feature vectors are the simple concatenation of these features. In our work, I focus on proposing a new model which can capture the share information between different views and use that information to perform feature selection for getting an optimized feature dimension simultaneously via topic modeling.

4.3 Problem Description

The research problem in this chapter is formally defined as follows: given a training set of image \ annotation pairs, learn a topic model and optimal feature dimension at the same time. While existing work focuses on designing new graph models,I study this modeling problem from the following angle: finding the optimal feature dimension for the topic model during the learning process. To achieve this goal, I proposed an procedure that updates feature dimension and model parameters iteratively.

Notations related to two-view LDA model of this chapter are described in the following.

symbol	Description
K	number of image-visual topics.
D,T,C	number of documents,unique textual words, unique image-visual words,respectively.
ϕ,	a $K \times C$ matrix indicating image visual topic word distribution.
θ,	a $D \times K$,matrix indicating image-visual topic proportions.
ψ	a $K \times T$ matrix indicating textual topic-word distribution.
$M_{d,t}$,$N^V_{d,c}$	number of textual words, image-visual words in the d-th document.
$M_{d,z}$	number of textual words in d-th document that are assigned to topic z.
$N^V_{d,k}$	number of image-visual words in d-th document that are assigned to topic k.

In my work, instead of learning topic model only on empirical features,I update the dimension of features in the meantime. The complete learning procedure is summarized as in Algorithm 2.

4.4 Solving the Proposed Model

To solve the above formulation, two parts need to be taken into consideration – parameter estimate for the LDA model and the feature selection optimization problem. We will explain the solution in detail in the following subsections.

4.4.1 Inference and Estimation of Two-view LDA

Exact probabilistic inference for LDA models are intractable; Therefore we need to approximate the posterior distribution over the latent variables given a particular image/annotation.Two major approach to do the inference are variational inference method and Gibbs sampling.

```
Algorithm 2: Learning the Two View LDA Model with Feature Selection
  Input      : Training image-annotation pairs (visual features and words)
  Parameters: Parameters for LDA models
  Output     : Trained LDA model
  Estimate of LDA model parameters;
  while not converge do
      Remove one dimension of the features which is not previously removed;
      Estimate LDA model parameters using Gibbs sampling method; Calculate
        perplexity;
      if perplexity increases then
          Adding back the removed feature dimension
      end
  end
```

We adopt Gibbs sampling to estimate the model parameters, due to its simplicity in deriving update rules and effectiveness in dealing with high-dimensional data. The basic idea of Gibbs sampling is to sequentially sample all variables from the targeted distribution when conditioned on the current values of all other variables and the data.

To sample for z_i, we condition on current value of all other variables, which leads to:

$$\begin{aligned} &P(z_i = k \mid W, T, Z_{-i}, Y) \\ &\propto \frac{N^V_{k,c,-i} + \beta_c}{N^V_k + C\beta - 1} \cdot (\frac{N^V_{d,k}}{N^V_{d,k-i}})^{Md,k} \cdot \frac{N^V_{d,k,-i} + \alpha_k}{N^V_d + K\alpha - 1} \end{aligned} \tag{4.1}$$

Similarly, we sample the latent topics y of the textual words based on the topic assignment of imagevisual words, which leads to:

$$\begin{aligned} &P(y_i = k \mid W, T, Z, Y_{-i}) \\ &\propto \frac{M_{k,t,-i} + \gamma}{M_k + T\gamma - 1} \cdot \frac{N_{d,k}}{N_d} \end{aligned} \tag{4.2}$$

Iterative execution of the above sampling rules until a steady state results allows us to obtain the values of the latent variables. Finally, we estimate the model parameters by the following equations:

$$\begin{aligned} \theta_{k,d} &= \frac{N^V_{k,d} + \alpha}{N^V_d + K\alpha} \\ \phi^V_{k,c} &= \frac{N^V_{k,c} + \beta}{N^V_k + C\beta} \\ \psi_{z,t} &= \frac{M_{z,t} + \gamma}{M_z + T\gamma} \end{aligned} \tag{4.3}$$

4.5 Experiments

In this section, we present experimental results based on a controlled synthetic dataset to show the performance of our proposed model and the comparison with existing LDA models without feature selection.

The reason we employ synthetic dataset here is that with synthetic dataset, we can control the ground truth as well as the generative process, which make the evaluation of model performance more straight forward than using real image dataset, which can be only evaluated by perplexity value or a secondary task.

We generate our synthetic dataset following the generative process described in Blei and Jordan (2003), and this part of data are referred as ”clean dataset”. We use 3-dimension feature to represent the visual features from an image, the size and dimension for the clean dataset is mentioned in Table 4.1.After we have the clean dataset, we manually add another 3 dimension noise to view 1. Now view 1 is noisy but view 2 is still clean. The underline assumption is that for real world data, visual features may contain noisy dimensions, and the text view are generally speaking free of noise.

We define 5 topics over both views. In the content of topic modeling, a topic is defined as a multinomial distribution over the target feature space. We also define 5 topic proportions, each topic proportion governed 100 documents. In reality, each document may have

	feature dimension	vocabulary size	training samples	testing samples
View 1	3	10	450	50
View 2	1	260	450	50

Table 4.1: Size and Dimension of Clean Synthetic Dataset.

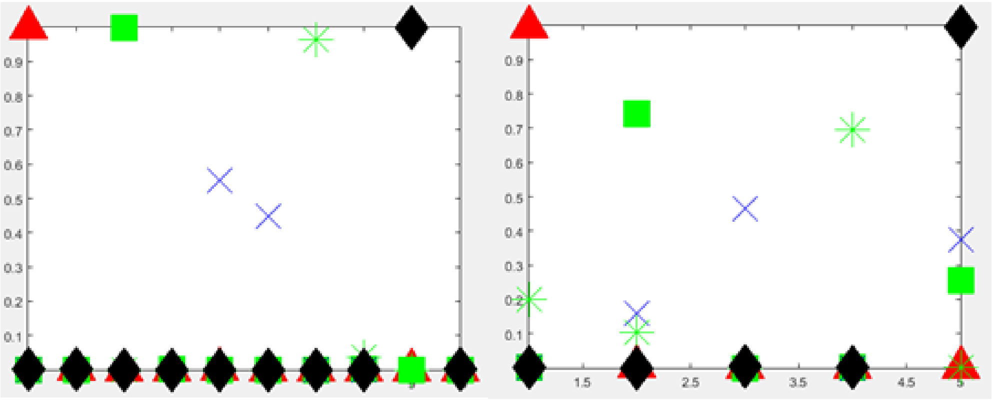


Figure 4.2: Left: 5 Topics Defined Over Visual Features; Right: 5 Topic Proportions Defined Over Topics.

different topic proportions, but for verification purpose, we simplify the settings.

In this experiment, we will see that the noise in feature space will affect the performance of topic modeling which proof the necessary of doing feature selection.

Figure 4.2 shows the ground truth of topics and topic proportions we used to generate clean synthetic data.

We now show the performance of two-view LDA model on different settings,eg. with or without feature selection. First we use our proposed model, the recovered topics and topic proportions are shown in Figure 4.3. It is worth noting that the order of the topics may change in the recovered versions, but it is easy to find its corresponding one in the ground truth. We can see that with feature selection embedded, the two-view LDA model

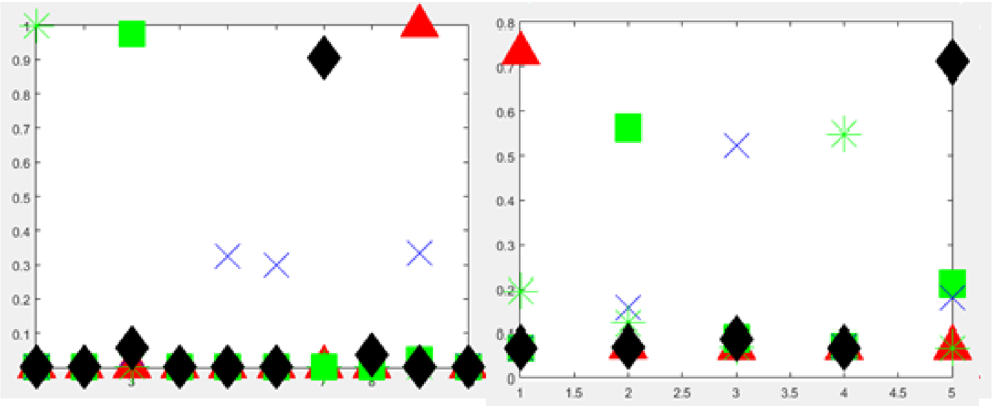


Figure 4.3: Left: Recovered 5 Topics; Right: Recovered 5 Topic Proportions.

can faithfully recover the topics and topic proportions that used as hyper parameters to generate the training documents.

Now we move on to evaluate the two-view LDA model without feature selection on noisy features. We do three experiments on different settings: 1. 3 clean dimensions+3 noise dimensions; 2. Remove one clean dimension; 3. Remove one noise dimension. The recovered results are shown in Figure 4.4, where the first column is recovered topic distributions and the second column is recovered topic proportions. The first setting reflects the general performance of LDA models on hand picked features and the latter two reflects the performance with some random dimension removed. We can see that among the three settings, the best performance is generated by removing one noise dimension. We also show the total KL distance between the ground truth and each of the recovered results in Figure 4.4 as well as the one with feature selection in Table 4.2. The recovered result generated by feature selection is closest to the ground truth.

4.6 Conclusion and Future Work

I present a new structure that combines topic model with feature selection component. The learning process iteratively update feature dimensions and model parameters. We cur-

	(a)	(b)	(c)	with feature selection
Topic distribution	1.6647	1.6216	0.6970	0.2053
Topic proportions	0.6471	0.7888	0.3418	0.2093

Table 4.2: KL Distances Between Different Results and the Ground Truth

rently only verified the idea on synthetic dataset. Future work will include evaluation on real image dataset with applications like tag prediction. There are several future directions can be considered: 1.Our current feature selection step requires exhausted search for an optimal solution, an more efficient way may which can simultaneously update the feature dimensions and model parameters is worth study; 2.Our current protocol of embedding feature selection is still an isolated process, a future direction would be to combine it with topic model would be using an objective function to link the feature selection result into the leaning process of topic model; 3.Another possible extension is to do feature selection among N views, N $\geq$ 3. The computational cost will be an issue for this extension as well as the complexity of the graph model.

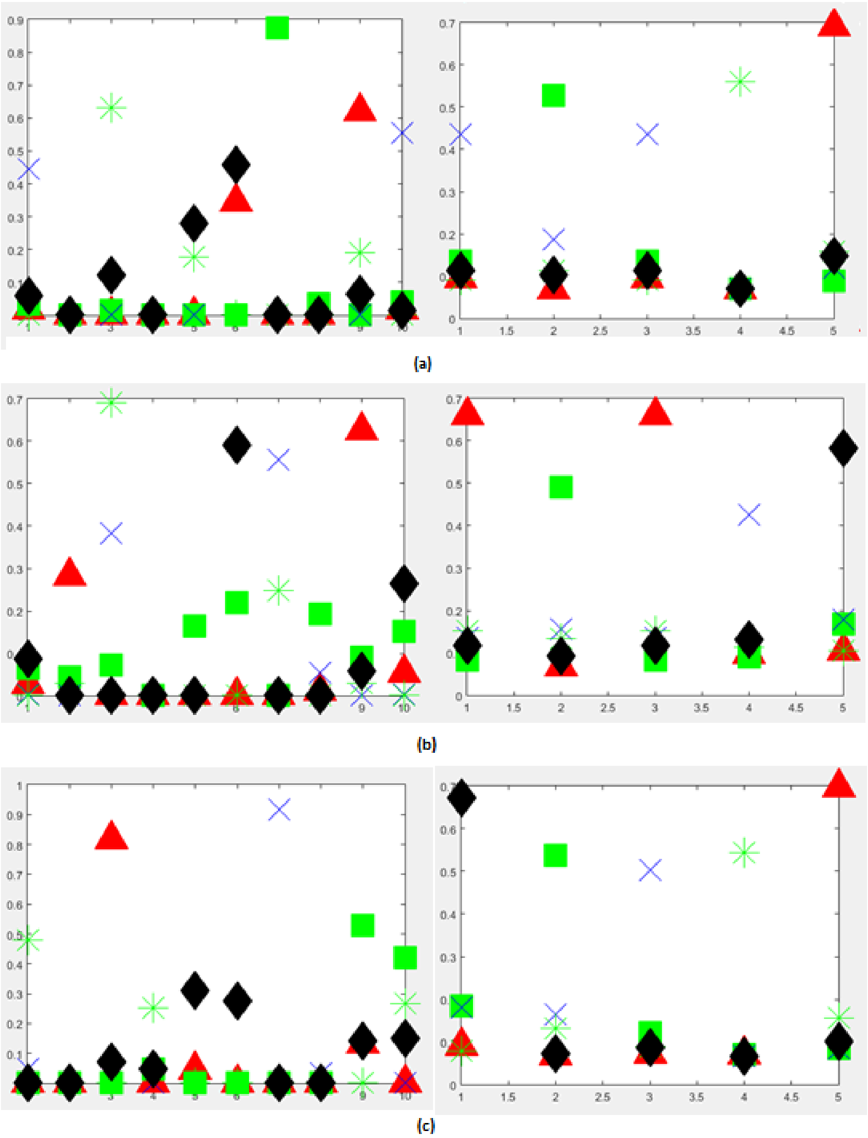


Figure 4.4: Recovered Results: (a) 3+3 Dimensions; (b) Remove One Clean Dimension; (c) Remove One Noisy Dimension.

Chapter 5

CONCLUSION AND FUTURE WORK

In this chapter, I summarize my major contributions in this dissertation and list some possible future extensions of my work.

5.1 Major Contributions

Visual processing, as an important field of study in the age of Internet and big data, involves many research tasks in Computer Vision. There are many challenges in this research field and I picked three issues which I feel are very important: handling scale differences in computer vision tasks like facial component detection and face retrieval; building efficient classifiers using partially labeled data and noisy data; and employing multi-modal model and feature selection to improve multi-view data analysis. For each solution, intensive experiments revealed that the proposed methods provide promising results.

Scale-insensitive Detection and Matching in Face Image Processing Contributions of this dissertation to scale-insensitive detection and matching in face image processing involves two works. First, I designed a quick and accurate scale-insensitive algorithm to detect eye centers and bounding box in wild images. As shown in many previous research works, eyes are among the most important facial recognition features of humans, and accurately locating eye positions can provide reliable foundations for many computer vision tasks. Secondly, I extended the scale-insensitive Eigen approach into a generalized Eigen problem setting. The extension makes it possible to accurately approximate the singular vector decomposition of linear transformed high order matrices in quick fashion. Several experiments were conducted and the results show the superiority of this proposed method.

Learning with partially labeled data and noisy data When using user-generated online media as the input to an analysis algorithm, it is difficult to avoid partially labeled or noisy data in many computer vision tasks. For this problem, I proposed two distinct algorithms. First, I proposed a dictionary-learning based approach with pair-wise constraints, which requires only relative labeling as training input. Furthermore, the constraints are designed to pick a portion of the image. Experiments on facial image datasets show that the proposed algorithm is robust to occlusions and noises. Secondly, I proposed a modified random forest structure that not only preserves the simplicity and efficiency of random forest, but also makes the original structure robust to class dependent label noise. This is the first work that links random forest to learning with noisy labels. Experiments on different datasets and various noise settings demonstrate the success of the proposed noise robust random forest.

Simultaneous clustering and feature seletion via LDA models Simultaneous clustering and feature selection via LDA models: LDA topic models have been intensively studied during the past decade. While much research has shown that LDA models are powerful in modeling multi-view data, few were concerned about the feature used in learning the model. I used synthetic datasets to show that noise in the feature space will hurt the performance of LDA models. Based on this observation, I bridged the gap by introducing feature selection into the learning framework. The complete procedure iteratively updates feature dimensions and model parameters.

5.2 Future work

In this section, I discuss some potential future extension of my research.

Preliminary exploration of embedded multi-view unsupervised feature selection Our proposed model involves an exhausted search for optimal feature dimensions. It is intuitive to explore a more efficient way of research. The embedded multi-view unsupervised feature selections are based on the following observations. Data in different views are intermittently divergent and similar, with many multimedia data collected from varying sources described from different views. For example, one image can be described by its different views such as color information, SIFT descriptor, word descriptions, and more. Intuitively, one could concatenate all the views into a single view before performing feature selection, as done in many previous works, or feature selection could take place before combining the results. Some disadvantages in these strategies include being unable to fully capture the relation among different views for coherent feature selection.

Assuming that we are given data having v representations (i.e., views). Let $\mathbf{X}^{(i)} = [\mathbf{x}^{(i)1}, \mathbf{x}^{(i)2}, ..., \mathbf{x}^{(i)n}] = [\mathbf{x}_1^{(i)\top}; \mathbf{x}_2^{(i)\top}, ..., \mathbf{x}_m^{(i)\top}]^\top \in \mathbb{R}^{m \times n}$ denotes the i-th view, m is the feature numbers and n is number of data samples; $\mathbf{x}^{(i)j}$ and $\mathbf{x}_k^{(i)}$ denote the j-th data sample and k-th feature in view $\mathbf{X}^{(i)}$, respectively. A cluster indicator is needed to control the clustering structure, in order to consider all views simultaneously in each iteration,and matrix $\mathbf{V}^{(i)}$ is the cluster indicator in each view. Based on the assumption that different views shared the same clustering structure, the superscript (i) from $\mathbf{V}$ can be dropped. By doing this all views are forced to use the same cluster indicator and $\mathbf{V}$ now acts as the bridge among views. The superscript (i) to α and β is for balancing different views. Finally the framework can be summarized as:

$$\begin{aligned} &\arg\min_{\mathbf{U}^{(i)},\mathbf{V}} \sum_{i=1}^{v} ||\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{V}^T||_F^2 + \sum_{i=1}^{v} \alpha^{(i)} ||\mathbf{U}^{(i)}||_{2,1} \\ &+ \sum_{i=1}^{v} \beta^{(i)} Tr(\mathbf{V}^\top \mathbf{L}^{(i)} \mathbf{V}), \quad s.t \quad \mathbf{V}^\top \mathbf{V} = \mathbf{I}, \mathbf{V} \geq 0 \end{aligned} \tag{5.1}$$

Since both $\mathbf{U}$ and $\mathbf{V}$ are sparse while $\mathbf{X}$ is not, the reconstruction error may easily dominate the objective function because of the $F-$norm is used. Thus, the loss function can be replaced by $l_{2,1}$-norm:

$$\begin{aligned}
&\arg\min_{\mathbf{U}^{(i)},\mathbf{V}} \sum_{i=1}^{v} ||\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{V}^{\top}||_{2,1} + \sum_{i=1}^{v} \alpha^{(i)} ||\mathbf{U}^{(i)}||_{2,1} \\
&+ \sum_{i=1}^{v} \beta^{(i)} Tr(\mathbf{V}^{T}\mathbf{L}^{(i)}\mathbf{V}), \quad s.t \quad \mathbf{V}^{\top}\mathbf{V} = \mathbf{I}, \mathbf{V} \geq 0
\end{aligned} \tag{5.2}$$

However, it is not easy to solve Eq. (5.2) directly since the objective function is not convex.This issue can be solved by updating $\mathbf{U}^{(i)}$ and $\mathbf{V}^{(i)}$ alternatively, then it becomes a convex optimization problem and Alternating Direction Method of Multiplier (ADMM) can be applied. By introducing two auxiliary variables $\mathbf{E}^{(i)} = \mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{V}^{T}$ and $\mathbf{Z} = \mathbf{V}$, the optimization problem becomes

$$\begin{aligned}
&\arg\min_{\mathbf{U}^{(i)},\mathbf{V},\mathbf{E}^{(i)},\mathbf{Z}} \sum_{i=1}^{v} ||\mathbf{E}^{(i)}||_{2,1} + \sum_{i=1}^{v} \alpha^{(i)} ||\mathbf{U}^{(i)}||_{2,1} + \sum_{i=1}^{v} \beta^{(i)} Tr(\mathbf{Z}^{T}\mathbf{L}^{(i)}\mathbf{V}) \\
&s.t \quad \mathbf{E}^{(i)} = \mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{V}^{\top}, \mathbf{Z} = \mathbf{V}, \mathbf{V}^{\top}\mathbf{V} = \mathbf{I}, \mathbf{Z} \geq 0
\end{aligned} \tag{5.3}$$

After adding Lagrangian multipliers $\mathbf{Y}_1$ and $\mathbf{Y}_2^{(i)}$, the final formulation is

$$\begin{aligned}
&\arg\min_{\mathbf{U}^{(i)},\mathbf{V},\mathbf{E}^{(i)},\mathbf{Z}} \sum_{i=1}^{v} ||\mathbf{E}^{(i)}||_{2,1} + \sum_{i=1}^{v} \alpha^{(i)} ||\mathbf{U}^{(i)}||_{2,1} \\
&+ \sum_{i=1}^{v} \beta^{(i)} Tr(\mathbf{Z}^{T}\mathbf{L}^{(i)}\mathbf{V}) + Tr(\mathbf{Y}_1^{\top}(\mathbf{Z} - \mathbf{V})) \\
&+ Tr(\mathbf{Y}_2^{(i)\top}(\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{V}^{\top} - \mathbf{E}^{(i)})) \\
&+ \frac{\mu}{2}(||\mathbf{Z} - \mathbf{V}||_F^2 + ||\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{V}^{\top} - \mathbf{E}^{(i)}||_F^2) \\
&s.t \quad \mathbf{V}^{T}\mathbf{V} = \mathbf{I}, \mathbf{Z} \geq 0
\end{aligned} \tag{5.4}$$

One simple way to initialize $\mathbf{V}$ and $\mathbf{U}^{(i)}$ is to set them to be all 0, but doing this will lead to slow convergence. Another way is to use k-means to obtain initial clusters as done in Wang *et al.* (2015). However, a sub-optimal solution might be possible due to the limitation of k-means. Considering the success of non-negative matrix factorization (NMF) for multi-view clustering as shown in Liu *et al.* (2013), it can be used to group $\mathbf{X}^{(i)}$ into k clusters as

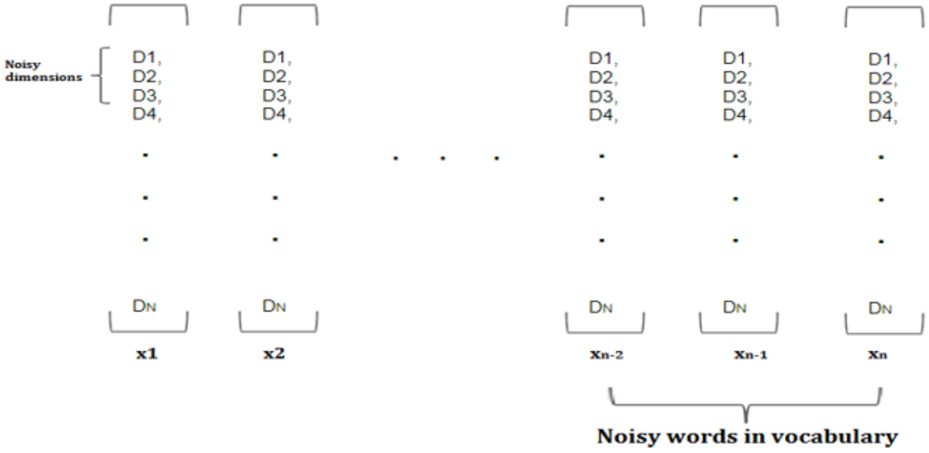


Figure 5.1: Optimize Dictionary Size and Feature Dimensions.

the initialization.This is a embedded unsupervised multi-view feature selection (EUMFS) formulation for the multi-view feature selection problem. This feature selection approach is a possible way to further improve the LDA model proposed in Chapter 4.

Finding the optimal dictionary size The term ”Visual word” is not so clear defined as text word. In general it refers to a small patch on the image (array of pixels) which can carry any kind of interesting information in any feature space (color changes, texture changes ...etc.).

Most existing methods do not consider the need of finding optimal visual dictionary size, but visual words exist in their feature space of continuous values, implying huge number of words and therefore a huge vocabulary. For effective modeling, it may be beneficial to reduce the number of visual words. As illustrate in Figure 5.1, optimizing visual dictionary size and visual feature dimension may be a good extension of our work.

REFERENCES

Adler, A. and M. E. Schuckers, “Comparing human and automatic face recognition performance”, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **37**, 5, 1248–1255 (2007).

Agrawal, S. and P. Khatri, “Facial expression detection techniques: based on viola and jones algorithm and principal component analysis”, in “Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on”, pp. 108–112 (IEEE, 2015).

Aharon, M., M. Elad and A. Bruckstein, “K -svd: An algorithm for designing overcomplete dictionaries for sparse representation”, IEEE Transactions on Signal Processing **54**, 11, 4311–4322 (2006).

Aharon, M., M. Elad and A. M. Bruckstein, “K-svd and its non-negative variant for dictionary design”, in “Optics & Photonics 2005”, pp. 591411–591411 (International Society for Optics and Photonics, 2005).

Belgiu, M. and L. Drăguţ, “Random forest in remote sensing: A review of applications and future directions”, ISPRS Journal of Photogrammetry and Remote Sensing **114**, 24–31 (2016).

Bellman, R. E., *Adaptive control processes: a guided tour*, vol. 2045 (Princeton university press, 2015).

Ben-David, S., J. Blitzer, K. Crammer, A. Kulesza, F. Pereira and J. W. Vaughan, “A theory of learning from different domains”, Machine learning **79**, 1-2, 151–175 (2010).

Berg, T. L., A. C. Berg, J. Edwards and D. A. Forsyth, “Who’s in the picture”, in “Advances in neural information processing systems”, pp. 137–144 (2005).

Blackard, J. A., *Comparison of neural networks and discriminant analysis in predicting forest cover types* (Colorado State University, 1998).

Blei, D. M. and M. I. Jordan, “Modeling annotated data”, in “Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval”, pp. 127–134 (ACM, 2003).

Blei, D. M., A. Y. Ng and M. I. Jordan, “Latent dirichlet allocation”, Journal of machine Learning research **3**, Jan, 993–1022 (2003).

Bosch, A., A. Zisserman and X. Munoz, “Image classification using random forests and ferns”, in “Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on”, pp. 1–8 (IEEE, 2007).

Brand, M., “Fast online svd revisions for lightweight recommender systems”, in “Proceedings of the 2003 SIAM International Conference on Data Mining”, pp. 37–46 (SIAM, 2003).

Breiman, L., “Random forests”, Machine Learning **45**, 1, URL https://doi.org/10.1023/A:1010933404324 (2001).

Burges, C. J., “Geometric methods for feature extraction and dimensional reduction”, in “Data mining and knowledge discovery handbook”, pp. 59–91 (Springer, 2005).

Burges, C. J. *et al.*, “Dimension reduction: A guided tour”, Foundations and Trends® in Machine Learning **2**, 4, 275–365 (2010).

Cai, D., C. Zhang and X. He, “Unsupervised feature selection for multi-cluster data”, in “Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 333–342 (ACM, 2010).

Chandrasekaran, V., S. Sanghavi, P. A. Parrilo and A. S. Willsky, “Rank-sparsity incoherence for matrix decomposition”, SIAM Journal on Optimization **21**, 2, 572–596 (2011).

Chen, T., H. M. SalahEldeen, X. He, M.-Y. Kan and D. Lu, “Velda: Relating an image tweet’s text and images.”, in “AAAI”, pp. 30–36 (2015).

Corrochano, E. B., *Handbook of Geometric Computing: Applications in Pattern Recognition, Computer Vision, Neuralcomputing, and Robotics* (Springer Science & Business Media, 2005).

Criminisi, A. and J. Shotton, *Decision forests for computer vision and medical image analysis* (Springer Science & Business Media, 2013).

Dai, W., Y. Fang and B. Hu, “Feature selection in interactive face retrieval”, in “Image and Signal Processing (CISP), 2011 4th International Congress on”, vol. 3, pp. 1358–1362 (IEEE, 2011).

De, A., A. Saha and M. Pal, “A human facial expression recognition model based on eigen face approach”, Procedia Computer Science **45**, 282–289 (2015).

De Bie, T., N. Cristianini and R. Rosipal, “Eigenproblems in pattern recognition”, in “Handbook of Geometric Computing”, pp. 129–167 (Springer, 2005).

De Lathauwer, L., B. De Moor and J. Vandewalle, “A multilinear singular value decomposition”, SIAM journal on Matrix Analysis and Applications **21**, 4, 1253–1278 (2000).

Del Río, S., V. López, J. M. Benítez and F. Herrera, “On the use of mapreduce for imbalanced big data using random forest”, Information Sciences **285**, 112–137 (2014).

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database”, in “Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on”, pp. 248–255 (IEEE, 2009).

Díaz-Uriarte, R. and S. A. De Andres, “Gene selection and classification of microarray data using random forest”, BMC bioinformatics **7**, 1, 3 (2006).

Ding, L. and A. M. Martinez, “Features versus context: An approach for precise and detailed detection and delineation of faces and facial features”, Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**, 11, 2022–2038 (2010).

Fowlkes, C., S. Belongie, F. Chung and J. Malik, “Spectral grouping using the nystrom method”, Pattern Analysis and Machine Intelligence, IEEE Transactions on **26**, 2, 214–225 (2004).

Frey, P. W. and D. J. Slate, “Letter recognition using holland-style adaptive classifiers”, Machine learning **6**, 2, 161–182 (1991).

Furl, N., A. OToole and P. Phillips, “Face recognition algorithms as models of the other race effect”, (2002).

Gan, L. and Q. Liu, “Eye detection based on rank order filter and projection function”, in “Computer Design and Applications (ICCDA), 2010 International Conference on”, vol. 1, pp. V1–642 (IEEE, 2010).

Ghosh, A., N. Manwani and P. Sastry, “On the robustness of decision tree learning under label noise”, in “Pacific-Asia Conference on Knowledge Discovery and Data Mining”, pp. 685–697 (Springer, 2017).

Goldstein, T. and S. Osher, “The split bregman method for l1-regularized problems”, SIAM Journal on Imaging Sciences **2**, 2, 323–343 (2009).

Gottumukkal, R. and V. K. Asari, “An improved face recognition technique based on modular pca approach”, Pattern Recognition Letters **25**, 4, 429–436 (2004).

Guillamet, D. and J. Vitria, “Classifying faces with nonnegative matrix factorization”, in “Proc. 5th Catalan conference for artificial intelligence”, pp. 24–31 (2002).

Guo, H., Z. Jiang and L. S. Davis, “Discriminative dictionary learning with pairwise constraints”, in “Asian Conference on Computer Vision(ACCV)”, pp. 328–342 (Springer, 2013).

Guo, Q., C. Zhang, Y. Zhang and H. Liu, “An efficient svd-based method for image denoising”, IEEE transactions on Circuits and Systems for Video Technology **26**, 5, 868–880 (2016).

Guyon, I. and A. Elisseeff, “An introduction to variable and feature selection”, The Journal of Machine Learning Research **3**, 1157–1182 (2003).

Hancock, P. J., V. Bruce and A. M. Burton, “Recognition of unfamiliar faces”, Trends in cognitive sciences **4**, 9, 330–337 (2000).

Hassaballah, M., T. Kanazawa and S. Ido, “Efficient eye detection method based on grey intensity variance and independent components analysis”, Computer Vision, IET **4**, 4, 261–271 (2010).

Hassner, T., S. Harel, E. Paz and R. Enbar, “Effective face frontalization in unconstrained images”, in “IEEE Conference on Computer Vision and Pattern Recognition (CVPR)”, (2015).

Hastie, T., R. Mazumder, J. D. Lee and R. Zadeh, “Matrix completion and low-rank svd via fast alternating least squares.”, Journal of Machine Learning Research **16**, 3367–3402 (2015).

He, X., D. Cai and P. Niyogi, “Laplacian score for feature selection”, in “Advances in neural information processing systems”, pp. 507–514 (2005).

Hu, B., B. Weng and S. Ruan, “Face recognition and retrieval based on feedback log information”, in “Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on”, vol. 1, pp. 578–584 (IEEE, 2012).

Huang, G. B., M. Ramesh, T. Berg and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments”, Tech. rep., Technical Report 07-49, University of Massachusetts, Amherst (2007).

Hull, J. J., “A database for handwritten text recognition research”, IEEE Transactions on pattern analysis and machine intelligence **16**, 5, 550–554 (1994).

Jain, V. and E. G. Learned-Miller, “Fddb: A benchmark for face detection in unconstrained settings”, UMass Amherst Technical Report (2010).

Jiang, Z., Z. Lin and L. S. Davis, “Learning a discriminative dictionary for sparse coding via label consistent k-svd”, in “IEEE Conference on Computer Vision and Pattern Recognition(CVPR)”, pp. 1697–1704 (2011).

Jin, C., S. M. Kakade and P. Netrapalli, “Provable efficient online matrix completion via non-convex stochastic gradient descent”, in “Advances in Neural Information Processing Systems”, pp. 4520–4528 (2016).

Jindal, I., M. S. Nokleby and X. Chen, “Learning deep networks from noisy labels with dropout regularization”, CoRR **abs/1705.03419**, URL `http://arxiv.org/abs/1705.03419` (2017).

Kontschieder, P., S. R. Bulò, A. Criminisi, P. Kohli, M. Pelillo and H. Bischof, “Context-sensitive decision forests for object detection”, in “Advances in neural information processing systems”, pp. 431–439 (2012).

Kontschieder, P., M. Fiterau, A. Criminisi and S. Rota Bulo, “Deep neural decision forests”, in “Proceedings of the IEEE International Conference on Computer Vision”, pp. 1467–1475 (2015).

Krizhevsky, A. and G. Hinton, “Learning multiple layers of features from tiny images”, (2009).

LeCun, Y., L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition”, Proceedings of the IEEE **86**, 11, 2278–2324 (1998).

Lee, D. D. and H. S. Seung, “Algorithms for non-negative matrix factorization”, in “Advances in Neural Information Processing Systems 13 (NIPS 2000)”, edited by T. Leen, T. Dietterich and V. Tresp, pp. 556–562 (MIT Press, 2001), URL `http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf`.

Lewin, C. and A. Herlitz, “Sex differences in face recognitionwomens faces make the difference”, Brain and cognition **50**, 1, 121–128 (2002).

Li, Z., Y. Yang, J. Liu, X. Zhou and H. Lu, “Unsupervised feature selection using nonnegative spectral analysis.”, in “AAAI”, (2012).

Liao, Y. and X. Lin, “Blind image restoration with eigen-face subspace”, Image Processing, IEEE Transactions on **14**, 11, 1766–1772 (2005).

Lindner, C., P. A. Bromiley, M. C. Ionita and T. F. Cootes, “Robust and accurate shape model matching using random forest regression-voting”, IEEE transactions on pattern analysis and machine intelligence **37**, 9, 1862–1874 (2015).

Liu, J., C. Wang, J. Gao and J. Han, “Multi-view clustering via joint nonnegative matrix factorization”, in “Proc. of SDM”, vol. 13, pp. 252–260 (SIAM, 2013).

Mahmud, F., S. Afroge, M. Al Mamun and A. Matin, “Pca and back-propagation neural network based face recognition system”, in “Computer and Information Technology (ICCIT), 2015 18th International Conference on”, pp. 582–587 (IEEE, 2015).

Manning, C. D., P. Raghavan and H. Schtze, “Relevance feedback and query expansion”, Introduction to Information Retrieval. Cambridge University Press, New York (2008).

Mardani, M., G. Mateos and G. B. Giannakis, “Subspace learning and imputation for streaming big data matrices and tensors”, IEEE Transactions on Signal Processing **63**, 10, 2663–2677 (2015).

Martinez, A. M., “The ar face database”, CVC Technical Report **24** (1998).

Mehta, R., J. Yuan and K. Egiazarian, “Face recognition using scale-adaptive directional and textural features”, Pattern Recognition **47**, 5, 1846–1858 (2014).

Menze, B. H., B. M. Kelm, D. N. Splitthoff, U. Koethe and F. A. Hamprecht, “On oblique random forests”, in “Joint European Conference on Machine Learning and Knowledge Discovery in Databases”, pp. 453–469 (Springer, 2011).

Mitra, P., C. Murthy and S. K. Pal, “Unsupervised feature selection using feature similarity”, Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**, 3, 301–312 (2002).

Montillo, A., J. Shotton, J. Winn, J. E. Iglesias, D. Metaxas and A. Criminisi, “Entangled decision forests and their application for semantic segmentation of ct images”, in “Biennial International Conference on Information Processing in Medical Imaging”, pp. 184–196 (Springer, 2011).

Montillo, A., J. Tu, J. Shotton, J. Winn, J. E. Iglesias, D. N. Metaxas and A. Criminisi, “Entanglement and differentiable information gain maximization”, in “Decision Forests for Computer Vision and Medical Image Analysis”, pp. 273–293 (Springer, 2013).

Natarajan, N., I. S. Dhillon, P. K. Ravikumar and A. Tewari, “Learning with noisy labels”, in “Advances in neural information processing systems”, pp. 1196–1204 (2013).

Pan, S. J., I. W. Tsang, J. T. Kwok and Q. Yang, “Domain adaptation via transfer component analysis”, IEEE Transactions on Neural Networks **22**, 2, 199–210 (2011).

Park, C. W., K. Park and Y. Moon, "Eye detection using eye filter and minimisation of nmf-based reconstruction error in facial image", Electronics letters **46**, 2, 130–132 (2010).

Park, U. and A. K. Jain, "Face matching and retrieval using soft biometrics", IEEE Transactions on Information Forensics and Security **5**, 3, 406–415 (2010).

Patrini, G., A. Rozza, A. Menon, R. Nock and L. Qu, "Making neural networks robust to label noise: a loss correction approach", arXiv preprint arXiv:1609.03683 (2016).

Poon, B., M. A. Amin and H. Yan, "Pca based face recognition and testing criteria", in "Machine Learning and Cybernetics, 2009 International Conference on", vol. 5, pp. 2945–2949 (IEEE, 2009).

Press, W. H., S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, "Numerical recipes in c: the art of scientific computing", Cambridge University Press, Cambridge, MA, **131**, 243–262 (1992).

Qiu, Q. and G. Sapiro, "Learning transformations for clustering and classification", The Journal of Machine Learning Research **16**, 1, 187–225 (2015).

Quach, K. G., C. N. Duong and T. D. Bui, "Sparse representation and low-rank approximation for robust face recognition", in "Pattern Recognition (ICPR), 2014 22nd International Conference on", pp. 1330–1335 (2014).

Quintiliano, P. and A. Santa-Rosa, "Face recognition based on eigeneyes", PATTERN RECOGNITION AND IMAGE ANALYSIS C/C OF RASPOZNAVANIYE OBRAZOV I ANALIZ IZOBRAZHENII **13**, 2, 335–338 (2003).

Rahmani, M. and G. Atia, "A subspace learning approach for high dimensional matrix decomposition with efficient column/row sampling", in "International Conference on Machine Learning", pp. 1206–1214 (2016).

Ren, J. and X. Jiang, "Eye detection based on rank order filter", in "Information, Communications and Signal Processing, 2009. ICICS 2009. 7th International Conference on", pp. 1–4 (IEEE, 2009).

Ren, S., X. Cao, Y. Wei and J. Sun, "Face alignment at 3000 fps via regressing local binary features", in "The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", (2014).

Ren, S., X. Cao, Y. Wei and J. Sun, "Global refinement of random forest", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 723–730 (2015).

Richmond, D. L., D. Kainmueller, M. Y. Yang, E. W. Myers and C. Rother, "Relating cascaded random forests to deep convolutional neural networks for semantic segmentation", arXiv preprint arXiv:1507.07583 (2015).

Rodriguez, J. J., L. I. Kuncheva and C. J. Alonso, "Rotation forest: A new classifier ensemble method", IEEE transactions on pattern analysis and machine intelligence **28**, 10, 1619–1630 (2006).

Schulter, S., P. Wohlhart, C. Leistner, A. Saffari, P. M. Roth and H. Bischof, "Alternating decision forests", in "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition", pp. 508–515 (2013).

Shamir, O., "A stochastic pca and svd algorithm with an exponential convergence rate", in "International Conference on Machine Learning", pp. 144–152 (2015).

Shenghua Gao, L.-T. C., Ivor Wai-Hung Tsang, "Kernel sparse representation for image classification and face recognition", in "European Conference on Computer Vision (ECCV)", pp. 1–14 (2010).

Shiau, Y. H. and C. C. Chen, "A sparse representation method with maximum probability of partial ranking for face recognition", in "2012 19th IEEE International Conference on Image Processing", pp. 1445–1448 (2012).

Shotton, J., T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook and R. Moore, "Real-time human pose recognition in parts from single depth images", Communications of the ACM **56**, 1, 116–124 (2013).

Skillicorn, D., *Understanding complex datasets: data mining with matrix decompositions* (CRC press, 2007).

Song, L., A. Smola, A. Gretton, K. M. Borgwardt and J. Bedo, "Supervised feature selection via dependence estimation", in "Proceedings of the 24th international conference on Machine learning", pp. 823–830 (ACM, 2007).

Sotoca, J. M. and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances", Pattern Recognition **43**, 6, 2068–2081 (2010).

Tang, X., Z. Ou, T. Su, H. Sun and P. Zhao, "Robust precise eye location by adaboost and svm techniques", in "Advances in Neural Networks–ISNN 2005", pp. 93–98 (Springer, 2005).

Theodorakopoulos, I., I. Rigas, G. Economou and S. Fotopoulos, "Face recognition via local sparse coding", in "2011 International Conference on Computer Vision", pp. 1647–1652 (2011).

Trefethen, L. N. and D. Bau III, *Numerical linear algebra*, vol. 50 (Siam, 1997).

Turk, M. and A. Pentland, "Eigenfaces for recognition", Journal of cognitive neuroscience **3**, 1, 71–86 (1991).

Van Loan, C. F., "Matrix computations (johns hopkins studies in mathematical sciences)", (1996).

Wang, P., M. B. Green, Q. Ji and J. Wayman, "Automatic eye detection and its validation", in "Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on", pp. 164–164 (IEEE, 2005a).

Wang, S., J. Tang and H. Liu, "Embedded unsupervised feature selection.", in "AAAI", pp. 470–476 (2015).

Wang, X. and X. Tang, “Random sampling for subspace face recognition”, International Journal of Computer Vision **70**, 1, 91–104 (2006).

Wang, Y., Y. Jia, C. Hu and M. Turk, “Non-negative matrix factorization framework for face recognition”, International Journal of Pattern Recognition and Artificial Intelligence **19**, 04, 495–511 (2005b).

Wang, Y., S. Wang, J. Tang, H. Liu and B. Li, “PPP: Joint pointwise and pairwise image label prediction”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, (2016).

Wang, Z., X. Xu and B. Li, “Bayesian tactile face”, in “Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on”, pp. 1–8 (IEEE, 2008).

Wright, J., A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, “Robust face recognition via sparse representation”, in “IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)”, vol. 31, Issue 2, pp. 210–227 (2009).

Wu, Z., Q. Ke, J. Sun and H.-Y. Shum, “Scalable face image retrieval with identity-based quantization and multireference reranking”, IEEE transactions on pattern analysis and machine intelligence **33**, 10, 1991–2001 (2011).

Xiao, T., T. Xia, Y. Yang, C. Huang and X. Wang, “Learning from massive noisy labeled data for image classification”, in “Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition”, pp. 2691–2699 (2015).

Xing, E. P., M. I. Jordan, R. M. Karp *et al.*, “Feature selection for high-dimensional genomic microarray data”, in “ICML”, vol. 1, pp. 601–608 (Citeseer, 2001).

Yang, M., D. Zhang, X. Feng and D. Zhang, “Fisher discrimination dictionary learning for sparse representation”, in “IEEE International Conference on Computer Vision (ICCV)”, pp. 543–550 (2011a).

Yang, M. and L. Zhang, “Gabor feature based sparse representation for face recognition with gabor occlusion dictionary”, in “European Conference on Computer Vision (ECCV)”, pp. 448–461 (2010).

Yang, Y., H. T. Shen, Z. Ma, Z. Huang and X. Zhou, “l2, 1-norm regularized discriminative feature selection for unsupervised learning”, in “IJCAI Proceedings-International Joint Conference on Artificial Intelligence”, vol. 22, p. 1589 (Citeseer, 2011b).

Yu, L. and H. Liu, “Feature selection for high-dimensional data: A fast correlation-based filter solution”, in “ICML”, vol. 3, pp. 856–863 (2003).

Zhang, Q. and B. Li, “Discriminative k-svd for dictionary learning in face recognition”, in “IEEE Conference on Computer Vision and Pattern Recognition (CVPR)”, pp. 2691–2698 (2010).

Zhang, T., “Solving large scale linear prediction problems using stochastic gradient descent algorithms”, in “Proceedings of the twenty-first international conference on Machine learning”, p. 116 (ACM, 2004).

Zhang, T., B. Fang, Y. Y. Tang, G. He and J. Wen, “Topology preserving non-negative matrix factorization for face recognition”, Image Processing, IEEE Transactions on **17**, 4, 574–584 (2008).

Zhao, Z. and H. Liu, “Spectral feature selection for supervised and unsupervised learning”, in “Proceedings of the 24th international conference on Machine learning”, pp. 1151–1157 (ACM, 2007).

Zhou, X., Y. Wang, P. Zhang and B. Li, “Scale-adaptive eigeneye for fast eye detection in wild web images”, in “Image Processing (ICIP), 2016 IEEE International Conference on”, pp. 2911–2915 (IEEE, 2016).

APPENDIX A

RELATED PUBLICATIONS

- Xu Zhou, Baoxin Li, “Retrieving Unfamiliar Faces: Towards Understanding Human Performance”, The 23rd ACM international conference on Multimedia (ACM MM), 2015.

- Xu Zhou, Yilin Wang, Peng Zhang, Baoxin Li, “Scale-adaptive Eigeneye for Fast Eye Detection in Wild Web Images”, IEEE International Conference on Image Processing (ICIP), 2016,

- Xu Zhou, Pak Lun Kevin Ding, Baoxin Li, “Non-Negative Dictionary Learning with Pairwise Partial Similarity Constraint”,IEEE International Conference on Multimedia and Expo (ICME),2017

- Xu Zhou, Pak Lun Kevin Ding, Baoxin Li,“Improving Robustness Of Random Forest Under Label Noise”, IEEE Winter Conference on Applications of Computer Vision (WACV), 2019

- Xu Zhou, Peng Zhang, Baoxin Li, “On linear transformed singular vector decomposition problem”. Pattern Recognition Letters, (under review)