ABSTRACT

Title of dissertation: MULTIMODAL LEARNING AND ITS APPLICATION TO MOBILE ACTIVE AUTHENTICATION

Heng Zhang, Doctor of Philosophy, May 2017

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

Mobile devices are becoming increasingly popular due to their flexibility and convenience in managing personal information such as bank accounts, profiles and passwords. With the increasing use of mobile devices comes the issue of security as the loss of a smartphone would compromise the personal information of the user.

Traditional methods for authenticating users on mobile devices are based on passwords or fingerprints. As long as mobile devices remain active, they do not incorporate any mechanisms for verifying if the user originally authenticated is still the user in control of the mobile device. Thus, unauthorized individuals may improperly obtain access to personal information of the user if a password is compromised or if a user does not exercise adequate vigilance after initial authentication on a device. To deal with this problem, active authentication systems have been proposed in which users are continuously monitored after the initial access to the mobile device [1]. Active authentication systems can capture users' data (facial image data, screen touch data, motion data, etc) through sensors (camera, touch screen, accelerometer, etc), extract features from different sensors' data, build classification models and

authenticate users via comparing additional sensor data against the models.

Mobile active authentication can be viewed as one application of the more general problem, namely, multimodal classification. The idea of multimodal classification is to utilize multiple sources (modalities) measuring the same instance to improve the overall performance compared to using a single source (modality). Multimodal classification also arises in many computer vision tasks such as image classification, RGBD object classification and scene recognition.

In this dissertation, we not only present methods and algorithms related to active authentication problems, but also propose multimodal recognition algorithms based on low-rank and joint sparse representations as well as multimodal metric learning algorithm to improve multimodal classification performance. The multimodal learning algorithms proposed in this dissertation make no assumption about the feature type or applications, thus they can be applied to various recognition tasks such as mobile active authentication, image classification and RGBD recognition.

First, we study the mobile active authentication problem by exploiting a dataset consisting of 50 users' face captured by the phone's frontal camera and screen touch data sensed by the screen for evaluating active authentication algorithms developed under this research. The dataset is named as UMD Active Authentication (UMDAA) dataset. Details on data preprocessing and feature extraction for touch data and face data are described respectively.

Second, we present an approach for active user authentication using screen touch gestures by building linear and kernelized dictionaries based on sparse representations and associated classifiers. Experiments using the screen touch data

components of UMDAA dataset as well as two other publicly available screen touch datasets show that the dictionary-based classification method compares favorably to those discussed in the literature. Experiments done using screen touch data collected in three different sessions show a drop in performance when the training and test data come from different sessions. This suggests a need for applying domain adaptation methods to further improve the performance of the classifiers.

Third, we propose a domain adaptive sparse representation-based classification method that learns projections of data in a space where the sparsity of data is maintained. We provide an efficient iterative procedure for solving the proposed optimization problem. One of the key features of the proposed method is that it is computationally efficient as learning is done in the lower-dimensional space. Various experiments on UMDAA dataset show that our method is able to capture the meaningful structure of data and can perform significantly better than many competitive domain adaptation algorithms.

Fourth, we propose low-rank and joint sparse representations-based multimodal recognition. Our formulations can be viewed as generalized versions of multivariate low-rank and sparse regression, where sparse and low-rank representations across all the modalities are imposed. One of our methods takes into account coupling information within different modalities simultaneously by enforcing the common low-rank and joint sparse representation among each modality's observations. We also modify our formulations by including an occlusion term that is assumed to be sparse. The alternating direction method of multipliers is proposed to efficiently solve the proposed optimization problems. Extensive experiments on

UMDAA dataset, WVU multimodal biometrics dataset and Pascal-Sentence image classification dataset show that that our methods provide better recognition performance than other feature-level fusion methods.

Finally, we propose a hierarchical multimodal metric learning algorithm for multimodal data in order to improve multimodal classification performance. We design metric for each modality as a product of two matrices: one matrix is modality specific, the other is enforced to be shared by all the modalities. The modality specific projection matrices capture the varying characteristics exhibited by multiple modalities and the common projection matrix establishes the relationship of the distance metrics corresponding to multiple modalities. The learned metrics significantly improves classification accuracy and experimental results of tagged image classification problem as well as various RGBD recognition problems show that the proposed algorithm outperforms existing learning algorithms based on multiple metrics as well as other state-of-the-art approaches tested on these datasets. Furthermore, we make the proposed multimodal metric learning algorithm non-linear by using kernel methods.

MULTIMODAL LEARNING AND ITS APPLICATION TO MOBILE ACTIVE AUTHENTICATION

by

Heng Zhang

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
May 2017

Advisory Committee:
Professor Rama Chellappa, Chair/Advisor
Professor Larry Davis
Professor Min Wu
Professor Vishal M. Patel, Rutgers University
Professor Ramani Duraiswami, Dean's Representative

Dedication

This dissertation is dedicated to God.

Acknowledgments

I owe my gratitude to all the people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I'd like to thank my advisor, Professor Rama Chellappa for giving me an invaluable opportunity to work with him on the challenging and interesting active authentication project. He has always made himself available for guidance and advice. He is hard-working, inspiring and humorous. It is my great honor and pleasure to learn from such a famous and influential researcher in the field of computer vision and pattern recognition.

I would also like to thank Professor Vishal M. Patel. I have been fortunate to work with him ever since I joined the active authentication team. We collaborated on all the research topics discussed in this dissertation. He is also caring and helpful just like a big brother for me. I enjoyed all the Friday lunches we had together when he was a research scientist at UMIACS and I wish him the best at Rutgers University.

Thanks are due to Professor Larry Davis, Professor Min Wu and Professor Ramani Duraiswami for agreeing to serve on my thesis committee and providing valuable feedbacks and advice to make this dissertation much better. I benefited from the suggestions and research ideas discussed by Professor Davis at active authentication meetings. ENEE630 taught by Professor Wu is the first course I took when I came to UMD. She made the course interesting and I learned a lot from her.

Professor Duraiswami has been very kind and helped me a lot during the final stage of my thesis defense.

The research experiences at Google gave me an opportunity to study active authentication problems at scale and explore methods and algorithms to make it work as a mobile application. The rigorous engineering practice and engineering culture at Google inspired me a lot. I would like to thank Dr. Deepak Chandra, Jagadish Agrawal and Brandon Barbello for hosting me and providing help in many ways.

My colleagues at Rama's group have enriched my graduate life. Specifically, Dr. Sumit Shekhar, Dr. Garrett Warnell, Dr. Ashish Shrivastava, Dr. Jie Ni, Dr. Jingjing Zheng, Dr. Huy Tho Ho, Mohammed E. Fathy, Howard Peng, Sayantan Sarkar and Swami Sankaranarayanan kindly discussed the research problems I had and provided useful advice. I would also like to acknowledge the help and support from staff members including Janice Perrone, Arlene Schenk, Bill Churma and Maria Hoo. This five-year journey at UMD would not be so enjoyable without the company of my friends including Zhihao, Hong, Zhe and many others. It is impossible to remember all, and I apologize to those I've inadvertently left out.

I owe my deepest thanks to my family. My parents have always stood by me and guided me through my life. My fiancee, Moyu has gone thourgh all the ups and downs with me. She encourages me to purse my dream and makes me a better person.

Lastly, thank you all and thank God!

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Motivation

Active user authentication on mobile devices has become an interesting research topic and attracted a lot of attention from both academia and industry. Active authentication is supposed to be performed by the mobile devices actively and continuously through sensing and analyzing users' physiological and behavioral data to decide whether the user is the trusted user or an impostor. With recent developments in hardware and software, current mobile devices, even the cheaper ones, can have many sensors including high resolution frontal and back cameras, touch screen, accelerometer and so on. These powerful sensors can acquire screen touch data, face images and so on.

The reasons why screen touch data and face data have been used to build mobile active authentication systems are 1) abundant screen touch data are available as long as users swipe on the screen and screen touch data can record detailed information which might be discriminative among users; 2) face recognition [2] is a relatively well-studied problem and several methods and techniques are at hand even though face images captured by mobile devices can exhibit different variations compared to the ones seen in traditional face recognition problems. However, screen

touch gestures, as a kind of human behavior, have a lot of intra-person variations and may change; face images captured by the mobile devices in unconstrained manner, can exhibit different poses, rotations, illuminations and partial faces. These challenges motivate us to study touch gesture and face-based active authentication by building efficient classification models for screen touch data and face images.

Mobile active authentication can be viewed as one application of multimodal classification which also arises in many computer vision tasks such as image classification, RGBD object classification and scene recognition. Rather than exploring one specific application, we study the more general multimodal recognition problems in order to robustly provide better performance than when just a single modality alone is used. However, the differences in features extracted from different modalities in terms of types and dimensions make the feature-level fusion non trivial. Simply concatenating feature vectors of multiple modalities and applying classic classification algorithms often yields poor performance and expensive computational cost since the dimension of the concatenated feature vector can be very large. The difficulty in fusing multiple feature vectors efficiently and effectively motivate us to explore robust multimodal fusion based on low-rank and joint sparse representations.

Metric learning algorithms can learn the Mahalanobis distance from data pairs and side information indicating the relationship of data pairs [3]. The learned distance can be better than the Euclidean distance for the original feature space and improve classification performance. While many classic metric learning algorithms in uni-modal setting are available, there are limited works on studying metric learning in multi-modal setting. The varying characteristics exhibited by multiple modalities

make it necessary to simultaneously learn the corresponding distance metrics. This motivate us to explore novel metric learning algorithms for multimodal data.

1.2 UMD Active Authentication Dataset

In order to facilitate mobile active authentication research, we built a dataset consisting of 50 users' face images and screen touch data over 3 sessions. The dataset is named as UMD Active Authentication(UMDAA) dataset.

In this section, we describe the details of the dataset we have collected using an iPhone 5s in an application environment. The users were asked to log in the data collection App and perform several tasks such as scrolling a document, viewing pictures, reading a long article etc. While users performed these tasks, their touch data sensed by the screen and face images acquired by the front-facing camera were simultaneously captured. Also, users need to perform these tasks in different sessions with different ambient conditions, namely in a well-lit room, in a dim-lit room, and in a room with natural daytime illumination. The goal was to simulate the real-world scenarios to study whether ambient changes can influence the captured face data and possibly users' touch behavior. During data collection, users were free to use the phone in either portrait mode or orientation mode and hold the phone in any position.

This dataset differs from other active authentication datasets including [4] and [5] in three aspects: a) data collection was done using the iOS platform, b) it is a multi-modal dataset consisting of face and touch data, c) data were collected

over three different sessions with different ambient conditions.

1.2.1 Data Collection App

The iPhone application for data collection consists of five different tasks described below. During each task, the application simultaneously records each users' face video from the front camera on the iPhone and the touch data sensed by the screen. Figure 1.1 shows the screen shots of the four different tasks for screen touch data collection.

Enrollment Task——An user would enroll face by turning his/her head to the left, then to the right, then up, and finally down while being recorded by the front-facing camera on the iPhone. Following the enrollment task, the user would perform four tasks with both face and screen touch data being recorded simultaneously. The four tasks are described as follows.

Scrolling Task——User would view a collection of images that are arranged horizontally and vertically. Each image would take up the whole screen and the user is required to swipe their finger on the screen left and right or up and down in order to navigate through the images.

Popup Task——Fifteen images are positioned off screen in such a way that only a segment of the image was shown. The user would then be required to drag the image and position it in the center of the iPhone to the best of their ability.

Picture Task——A large poster-like image displays 72 cars with different colors in a 12 by 6 table. Only a few cars could be seen at any given time on the screen.

The user was then asked to count the number of cars that were of the color selected by the test proctor. The user was then required to scroll through the entire image in order to provide the correct number.

Documents Task—This task contains a PDF of a research paper which is 12 pages long. The user was asked to count the number of items indicated by the test proctor such as figure, tables etc.

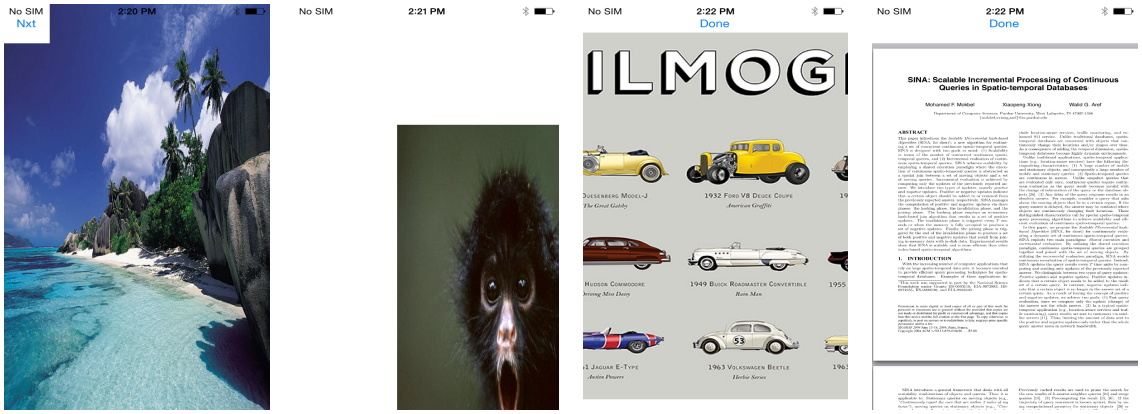


Figure 1.1: Screen shots of the App for data collection. These four pictures from left to right are screen shots of Scrolling Task, Popup Task, Picture Task and Document Task respectively.

1.2.2 Data Visualization

On average it took about 30 seconds to 2 minutes to collect facial and touch data per task per session. The dataset consists of 50 users with 43 male users and 7 female users. All 50 users used the phone in the portrait mode and only one user also used the phone in the landscape mode. In total, there are 750 videos recording

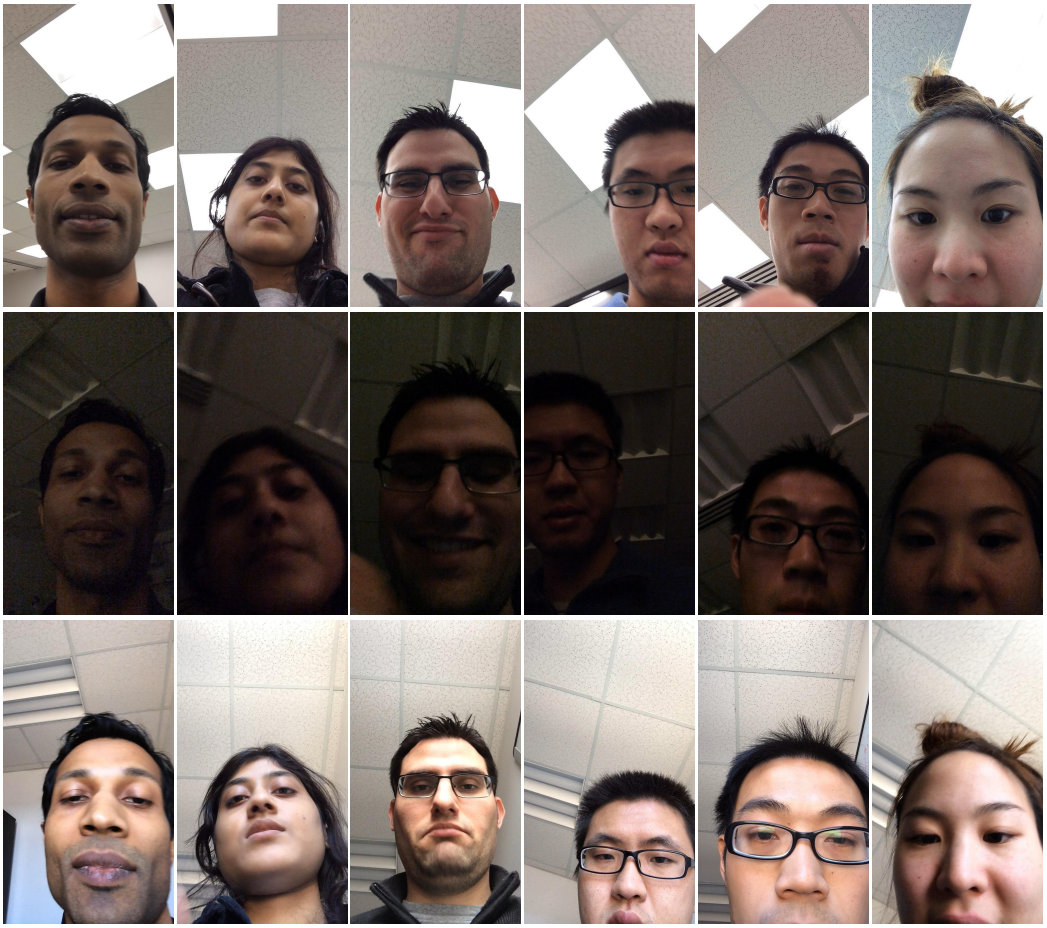


Figure 1.2: Examples of face images in UMDAA dataset. Each row shows face images collected from a mobile device in a particular ambient condition. Images in each column correspond to the same individual.

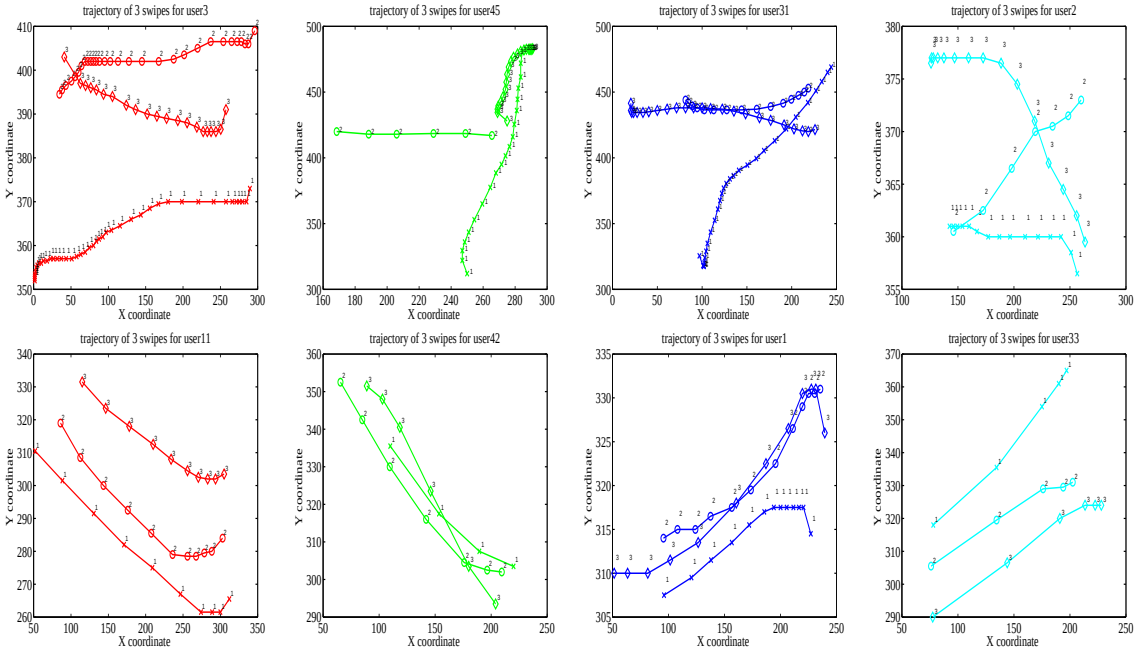


Figure 1.3: Samples of screen touch data in UMDAA dataset. First and second rows respectively show touch data corresponding to four different individuals performing the same task. The figure is best viewed in color and 200% zoom in.

facial data and 600 txt files recording screen touch data with about 15490 touch gestures.

Since facial video data were collected in an unconstrained manner, many faces exhibit different poses, rotations and illuminations. In particular, partial faces are common in this dataset. Also, as users are free to swipe on the screen in any way they prefer, intra-user variations can be large. Figures 1.2 shows sample face images from this dataset. Each row shows images from a particular ambient condition. It can be seen that the images from different ambient conditions show very different characteristics. Figure 1.3 shows samples of touch data in this dataset. It is interesting to observe that even for the same task touch data of different users show significant differences.

1.2.3 Preprocessing and Feature Extraction

As this dataset consists of two modalities, we perform preprocessing and feature extraction for face and screen touch data separately.

1.2.3.1 Facial data

For the face data, we first detect the landmarks of the face images frame by frame from the videos using the tree-based landmarks detector [6]. These detected landmarks are shown in the second row of Figure 1.4. We then crop and align the faces using the method described in [7] based on the landmarks' locations. We then apply the illumination normalization method described in [8] to the cropped

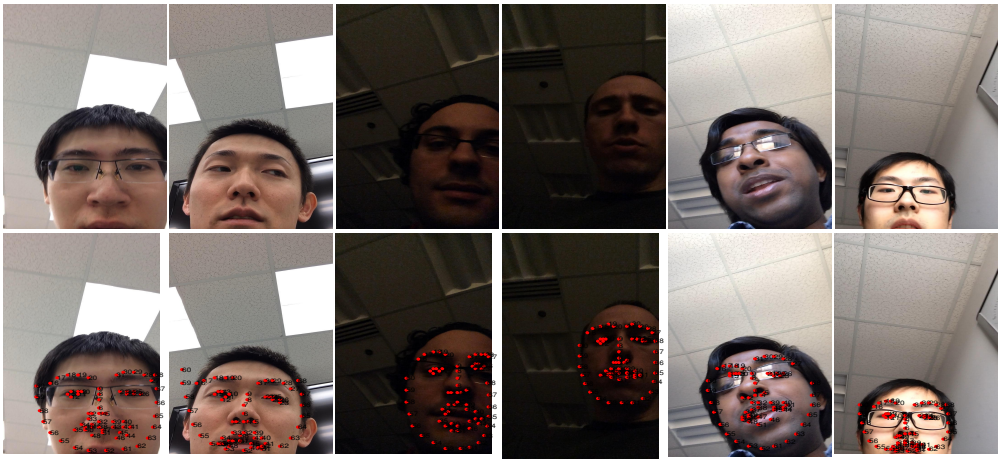


Figure 1.4: First row: Example faces in this dataset. Second row: Detected landmarks on the images shown on the first row.

face images. Finally, the face images are rescaled to dimension $192 \times 168 \times 3$ and converted to grayscale images. After preprocessing, we downsample the preprocessed face images to 24 by 21 and simply used the whole image as a feature vector of dimension 504.

1.2.3.2 Touch Data

Every swipe on the screen is a sequence of touch data when the finger is in touch with the screen of the mobile phone. Every swipe $\mathbf{S}$ is encoded as a sequence of vectors

$$\mathbf{s}_i = (x_i, y_i, t_i, A_i, o_i^{ph}),$$

$i \in \{1, \cdots, N_c\}$ where x_i, y_i are the location points, t_i is the time stamp, A_i is the area occluded by the finger and o_i^{ph} is the orientation of the phone (e.g. landscape or portrait). Given these touch data, we extracted a 27-dimensional feature vector for

FeatureID	Description
feature 1	inter-stroke time
feature 2	stroke duration
feature 3	start x
feature 4	start y
feature 5	stop x
feature 6	stop y
feature 7	direct end-to-end distance
feature 8	mean resultant length
feature 9	up/down/left/right flag
feature 10	direction of end-to-end line
feature 11	20%-perc. pairwise velocity
feature 12	50%-perc. pairwise velocity
feature 13	80%-perc. pairwise velocity
feature 14	20%-perc. pairwise acceleration
feature 15	50%-perc. pairwise acceleration
feature 16	80%-perc. pairwise acceleration
feature 17	median velocity at last 3 points
feature 18	largest deviation from end-to-end line
feature 19	20%-perc. dev. from end-to-end line
feature 20	50%-perc. dev. from end-to-end line
feature 21	80%-perc. dev. from end-to-end line
feature 22	average direction
feature 23	length of trajectory
feature 24	ratio end-to-end dist and length of trajectory
feature 25	average velocity
feature 26	median acceleration at first 5 points
feature 27	mid-stroke area covered

Table 1.1: Description of the 27-dimensional feature vector.

every single stroke in our dataset using the method described in [5]. These features are summarized in Table 1.1.

Note that for the Touchalytics [5] and the BTAS 2013 dataset [4], we extracted 28 features from each swipe. Additional feature for these datasets corresponds to the mid-stroke pressure. The new dataset described in Section 3 was collected using an iPhone 5s and it does not allow one to capture the pressure information. Whereas, the Touchalytics dataset and the BTAS 2013 dataset, were collected using Android phones which allow them to collect the pressure information.

1.3 Proposed Algorithms and Contributions

1.3.1 Touch Gesture-Based Active User Authentication Using Dictionaries

Screen touch gesture has been shown to be a promising modality for touch-based active authentication of users of mobile devices. We present an approach for active user authentication using screen touch gestures by building kernelized dictionaries based on sparse representations and associated classifiers.

This work makes the following contributions:

- We propose kernel dictionary learning-based methods for touch gesture-based active user authentication.
- We point out the domain shift issue for touch-based active authentication and suggest future research work in this area to address these challenges.

1.3.2 Domain Adaptive Sparse Representation-Based Classification

We propose Domain Adaptive Sparse Representation-Based Classification which combines subspace learning and sparse representation-based classification (SRC) [9] and attempts to mitigate the domain shift. The proposed formulations learn projections of data in different domains in a way that preserves the sparse structure of data in the low-dimensional space. We develop an efficient optimization method based on Alternating Direction Method of Multipliers (ADMM) and the Method of Splitting Orthogonality Constraints (SOC) for solving the resulting optimization problem.

This work makes the following contributions:

- A sparse representation-based classification algorithm is proposed for domain adaptation.
- An efficient iterative method based on the ADMM and the method of SOC is derived for solving the resulting optimization problem.
- The effectiveness of the proposed domain adaptation approach is demonstrated through comparisons with other recently proposed state-of-the-art domain adaptation methods on faces images and screen touch data of UMDAA dataset.

1.3.3 Low-Rank and Joint Sparse Representations for Multimodal Recognition

We propose multimodal feature-level fusion methods by simultaneously enforcing low-rank and joint sparsity constraints across representations corresponding to multiple modalities. The proposed method is a general formulation for multimodal fusion problems where different representations (sparse and low-rank) are simultaneously sought for improved multimodal fusion. Efficient optimization algorithms using ADMM is derived to solve the proposed optimization problems.

This work makes the following contributions:

- A general formulation based on low-rank and joint sparse representations is proposed for multimodal recognition.
- An extended formulation based on common sparse and low-rank representation is proposed to robustly leverage the correlation and coupling information across the modalities especially when the performance of each modality differs a lot.
- We evaluate our method on various multimodal recognition problems such as active authentication [10], [11] multi-biometrics recognition [12], and image recognition [13].

1.3.4 Hierarchical Multimodal Metric Learning for multimodal Classification

We propose a Hierarchical Multimodal Metric Learning (HM3L) algorithm which fully exploits the relationships among the different metrics of different modalities. In our formulation, the metric of each modality is constructed through the multiplication of modality specific part representing appropriate subspace and a common part ($p.s.d$ matrix) shared by all the metrics. Furthermore, The kernelization of the proposed algorithm leads to Kernelized Hierarchical Multimodal Metric Learning (KHM3L) algorithm and can be applied to classification problems in which decision boundary is complex.

This work makes the following contributions:

- A noval Mahalanobis metric learning algorithm for multimodal data is proposed by factoring the distance metric of each modality into the product of a modality-specific projection and a common projection shared across all metrics.

- Kernelization of the proposed HM3L is derived to learn metrics for multimodal data in kernel space.

- We evaluate the proposed method on four publicly available multimodal datasets about RGB-D recognition and tagged image classification. We obtain the state-of-the-art results with 89.2% object category recognition accuracy on the multi-view RGB-D dataset [14] and 52.3% scene category recognition ac-

curacy on SUN RGB-D dataset [15].

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we build touch gesture dictionaries as user biometric templates to perform active authentication. Then, we propose in Chapter 3, a domain adaptive sparse representation-based classification algorithm to mitigate the domain shift issues for face and screen touch data. Next, in Chapter 4, we formulate a multimodal recognition algorithm using low-rank and joint sparse representations in order to perform efficient and robust feature level fusion. In Chapter 5, we propose a hierarchical multimodal metric learning (HM3L) algorithm and its kernelized extension (KHM3L) to improve multimodal classification performance. Finally, in Chapter 6, we conclude this dissertation with a summary and discussion of future research.

Chapter 2: Touch Gesture-Based Active User Authentication Using Dictionaries

2.1 Introduction

Screen touch gestures, as a kind of behavioral biometric, are basically the way users swipe their fingers on the screen of their mobile devices. They have been used to continuously authenticate users while users perform basic operations on the phone [5], [4], [16], [17]. In these methods, a behavioral feature vector is extracted from the recorded screen touch data and a discriminative classifier like an SVM classifier or a Nearest Neighbor classifier is trained on these extracted features for authentication. These works have demonstrated that touch gestures can be used as a promising biometric for active user authentication of mobile devices in the future.

In recent years, sparse representation and dictionary learning based methods have produced state-of-the-art results in many physiological biometrics recognition problems such as face recognition [18] and iris recognition [19]. These methods assume that given sufficient training samples of certain class, any new test sample that belongs to the same class will lie approximately in the linear or nonlinear span of the training samples from that class. We assume that this assumption is also

valid for behavioral biometric, like screen touch gestures.

Kernel sparse coding [20] and kernel dictionary learning [21] have been proposed and applied for image classification and face recognition. In this chapter, we study the effectiveness of kernel dictionary learning-based methods in recognizing screen touch gestures for user authentication. Our method builds dictionaries for users, which can be viewed as biometric templates of users and are more suitable to be incorporated into a biometric system to authenticate users actively and continuously. Application of kernel dictionary learning for touch gesture recognition and achieving very promising performance are the primary goals of this work.

The rest of this chapter is organized as follows. Section 2.2 describes sparse representation and dictionary learning-based methods for screen touch gesture recognition. Experimental results on screen touch component of this new dataset as well as on two other publicly available screen touch datasets are presented in Section 2.3. Finally, Section 2.4 presents a brief summary and discussion.

2.2 Sparse Representation and Dictionary Learning based Classification

2.2.1 Sparse Representation-based Classification(SRC)

Suppose that we are given C distinct classes and a set of N_c training samples per class. Let $\mathbf{Y}_c = [\mathbf{y}_1^c, \ldots, \mathbf{y}_{N_c}^c] \in \mathbb{R}^{d \times N_c}$ be the matrix of training samples from the cth class. Define a matrix $\mathbf{Y}$ as the concatenation of training samples from all

the classes

$$\mathbf{Y} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_C] \in \mathbb{R}^{d \times N} = [\mathbf{y}_1^1, \ldots, \mathbf{y}_{N_1}^1 | \mathbf{y}_1^2, \ldots, \mathbf{y}_{N_2}^2 | \ldots\ldots | \mathbf{y}_1^C, \ldots, \mathbf{y}_{N_C}^C],$$

where $N = \sum_c N_c$. We consider an observation vector $\mathbf{y}_t \in \mathbb{R}^d$ of unknown class as a linear combination of the training vectors as

$$\mathbf{y}_t = \sum_{c=1}^{C} \sum_{i=1}^{N_c} x_i^c \mathbf{y}_i^c \tag{2.1}$$

with coefficients $x_i^c \in \mathbb{R}$. Equation (2.1) can be more compactly written as

$$\mathbf{y}_t = \mathbf{Y}\mathbf{x}, \tag{2.2}$$

where $\mathbf{x} = [x_1^1, \ldots, x_{N_1}^1 | x_1^2, \ldots, x_{N_2}^2 | \ldots | x_1^C, \ldots, x_{N_C}^C]^T$.

One can make an assumption that given sufficient training samples of the cth class, $\mathbf{Y}_c$, any new test sample $\mathbf{y}_t \in \mathbb{R}^d$ that belongs to the same class will approximately lie in the linear span of the training samples from the class c. This implies that most of the coefficients not associated with class c will be close to zero. As a result, assuming that observations are noisy, one can recover this sparse vector by solving the following optimization problem,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \tag{2.3}$$

$$s.t.\ \|\mathbf{y}_t - \mathbf{Y}\mathbf{x}\|_2 \leq \epsilon$$

or equivalently the following formulation,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{y}_t - \mathbf{Y}\mathbf{x}\|_2 + \lambda \|\mathbf{x}\|_0, \tag{2.4}$$

where λ is a parameter and $\|\cdot\|_p$ for $0 < p < \infty$ is the ℓ_p-norm defined as

$$\|\mathbf{x}\|_p = \left(\sum_{j=1}^{d} |x_j|^p \right)^{\frac{1}{p}}.$$

The sparse code $\mathbf{x}_t$ can be solved using Orthogonal Matching Pursuit (OMP) algorithm [22]. Then the class of $\mathbf{y}_t$ can be determined by computing the following error for each class,

$$e_c = \|\mathbf{y}_t - \mathbf{Y}_c \mathbf{x}_t^c\|_2, \tag{2.5}$$

where, $\mathbf{x}_t^c$ is the part of coefficient vector $\mathbf{x}_t$ that corresponds to $\mathbf{Y}_c$. Finally, the class c^* that is assigned to the test sample $\mathbf{y}_t$, can be declared as the one that produces the smallest approximation error [18]

$$c^* = \arg\min_c e_c = \arg\min_c \|\mathbf{y}_t - \mathbf{Y}_c \mathbf{x}_t^c\|_2. \tag{2.6}$$

2.2.2 Kernel Sparse Representation-based Classification (KSRC)

In kernel SRC, the idea is to map data in the high dimensional feature space and solve (2.4) using the kernel trick [20] [21]. This allows one to deal with data which are not linearly separable in the original space [23]. Let $\mathbf{\Phi} : \mathbb{R}^d \rightarrow G$ be a non-linear mapping from the d-dimensional space into a dot product space G. A non-linear SRC can be performed by solving the following optimization problem,

$$\mathbf{x}_t = \arg\min_{\mathbf{x}} \|\mathbf{\Phi}(\mathbf{y}_t) - \mathbf{\Phi}(\mathbf{Y})\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_0, \tag{2.7}$$

where

$$\mathbf{\Phi}(\mathbf{Y}) \triangleq [\mathbf{\Phi}(\mathbf{y}_1^1), \cdots, \mathbf{\Phi}(\mathbf{y}_{N_1}^1) | \cdots | \mathbf{\Phi}(\mathbf{y}_1^C), \cdots, \mathbf{\Phi}(\mathbf{y}_{N_C}^C)].$$

Denote the first term of (2.7) by $\mathcal{E}_\kappa$ as follows

$$\begin{aligned}\mathcal{E}_\kappa(\mathbf{x};\mathbf{Y},\mathbf{y}_t) &= \|\mathbf{\Phi}(\mathbf{y}_t)-\mathbf{\Phi}(\mathbf{Y})\mathbf{x}\|_2^2 \\ &= \mathbf{\Phi}(\mathbf{y}_t)^T\mathbf{\Phi}(\mathbf{y}_t)+\mathbf{x}^T\mathbf{\Phi}(\mathbf{Y})^T\mathbf{\Phi}(\mathbf{Y})\mathbf{x}-2\mathbf{\Phi}(\mathbf{y}_t)^T\mathbf{\Phi}(\mathbf{Y})\mathbf{x} \\ &= \kappa(\mathbf{y}_t,\mathbf{y}_t)+\mathbf{x}^T\boldsymbol{\mathcal{K}}(\mathbf{Y},\mathbf{Y})\mathbf{x}-2\boldsymbol{\mathcal{K}}(\mathbf{y}_t,\mathbf{Y})\mathbf{x},\end{aligned}$$

where $\boldsymbol{\mathcal{K}}(\mathbf{Y},\mathbf{Y})\in\mathbb{R}^{N\times N}$ is a positive semidefinite kernel Gram matrix whose elements are computed as

$$[\boldsymbol{\mathcal{K}}(\mathbf{Y},\mathbf{Y})]_{i,j}=[\langle\mathbf{\Phi}(\mathbf{Y}),\mathbf{\Phi}(\mathbf{Y})\rangle]_{i,j}=\mathbf{\Phi}(\mathbf{y}_i)^T\mathbf{\Phi}(\mathbf{y}_j)=\kappa(\mathbf{y}_i,\mathbf{y}_j),$$

and

$$\boldsymbol{\mathcal{K}}(\mathbf{y}_t,\mathbf{Y})\triangleq[\kappa(\mathbf{y}_t,\mathbf{y}_1),\kappa(\mathbf{y}_t,\mathbf{y}_2),\cdots,\kappa(\mathbf{y}_t,\mathbf{y}_N)]\in\mathbb{R}^{1\times N}, \tag{2.8}$$

where $\kappa:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ is the kernel function.

Note that the computation of $\boldsymbol{\mathcal{K}}$ only requires dot products. Therefore, we are able to employ Mercer kernel functions to compute these dot products without carrying out the mapping $\mathbf{\Phi}$. Some commonly used kernels include polynomial kernels

$$\kappa(\mathbf{x},\mathbf{y})=\langle(\mathbf{x},\mathbf{y}\rangle+a)^b$$

and Gaussian kernels

$$\kappa(\mathbf{x},\mathbf{y})=\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{c}\right),$$

where a,b and c are the parameters.

With the above definitions, the kernel version of the SRC optimization problem in (2.4) can be written as,

$$\mathbf{x}_t=\arg\min_{\mathbf{x}}\mathcal{E}_\kappa(\mathbf{x};\mathbf{Y},\mathbf{y}_t)+\lambda\|\mathbf{x}\|_0. \tag{2.9}$$

One can solve the optimization problem (2.9) by the kernel orthogonal matching pursuit algorithm [21].

2.2.3 Dictionary Learning-based Classification

Rather than finding a sparse representation based on training samples as is done in SRC, C touch specific dictionaries can be trained by solving the following optimization problem for $i = 1, \cdots, C$. The following optimization problem can be efficiently solved by the KSVD algorithm [24].

$$(\hat{\mathbf{D}}_i, \hat{\mathbf{X}}_i) = \arg\min_{\mathbf{D}_i, \mathbf{X}_i} \|\mathbf{Y}_i - \mathbf{D}_i\mathbf{X}_i\|_F^2 \tag{2.10}$$

$$s.t.\ \|\mathbf{x}_j\|_0 \leq T_0\ \forall j$$

Given a test sample $\mathbf{y}_t$, first we compute its sparse codes $\mathbf{x}_i$ with respect to each $\mathbf{D}_i$ using OMP algorithm [22], and then compute reconstruction error

$$\mathbf{r}_i(\mathbf{y}_t) = \|\mathbf{y}_t - \mathbf{D}_i\mathbf{x}_i\|_F^2$$

.

Since the KSVD algorithm finds the dictionary, $\mathbf{D}_i$, that leads to the best representation for each examples in $\mathbf{Y}_i$, one can expect $\mathbf{r}_i(\mathbf{y}_t)$ to be small if $\mathbf{y}_t$ were to belong to the i^{th} class and large for the other classes. Based on this, one can classify $\mathbf{y}_t$ by finding the class corresponding to the lowest reconstruction error. Note that similar methods have been used for face biometric in [25].

2.2.4 Kernel Dictionary Learning-based Classification

Linear dictionary learning model (2.10) can be made non-linear so that non-linearity in the data can be handled better [21]. Kernel dictionary learning optimization can be formulated as follows

$$(\hat{\mathbf{A}}_i, \hat{\mathbf{X}}_i) = \arg\min_{\mathbf{A}_i, \mathbf{X}_i} \|\mathbf{\Phi}(\mathbf{Y}_i) - \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i\mathbf{X}_i\|_F^2 \tag{2.11}$$
$$s.t.\ \|\mathbf{x}_j\|_0 \leq T_0\ \forall j,$$

where the dictionary in the feature space [21] is modeled as follows

$$\mathbf{D} = \mathbf{\Phi}(\mathbf{Y})\mathbf{A}, \tag{2.12}$$

where $\mathbf{A}$ is a coefficient matrix. This model provides adaptivity via modification of matrix $\mathbf{A}$. After some algebraic manipulations, the cost function in (2.11) can be written as

$$\|\mathbf{\Phi}(\mathbf{Y}_i) - \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i\mathbf{X}_i\|_F^2 = tr((\mathbf{I} - \mathbf{A}_i)^T \boldsymbol{\mathcal{K}}(\mathbf{Y}_i, \mathbf{Y}_i)(\mathbf{I} - \mathbf{A}_i)). \tag{2.13}$$

This problem can be solved using Kernel KSVD (KKSVD) algorithm [21] which applies sparse coding in kernel space and dictionary update in kernel space.

Let $\mathbf{D}_i = \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i$ denote the learned kernel dictionary for each class, where $i \in \{1, \cdots, C\}$. Given a test sample $\mathbf{y}_t$, first perform kernel OMP separately for each $\mathbf{D}_i$ to obtain the sparse code $\mathbf{x}_i$. Similarly, the test sample is assigned to the class that gives the smallest reconstruction error. Reconstruction error $\mathbf{r}_i(\mathbf{y}_t)$

$(i \in [1, \cdots, C])$ is computed as

$$\begin{aligned} \mathbf{r}_i(\mathbf{y}_t) &= \|\mathbf{\Phi}(\mathbf{y}_t) - \mathbf{\Phi}(\mathbf{Y}_i)\mathbf{A}_i\mathbf{x}_i\|_F^2 \\ &= \kappa(\mathbf{y}_t, \mathbf{y}_t) - 2\boldsymbol{\mathcal{K}}(\mathbf{y}_t, \mathbf{Y}_i)\mathbf{A}_i\mathbf{x}_i + \mathbf{x}_i^T\mathbf{A}_i^T\boldsymbol{\mathcal{K}}(\mathbf{Y}_i, \mathbf{Y}_i)\mathbf{A}_i\mathbf{x}_i \end{aligned} \tag{2.14}$$

2.3 Experimental Results On Touch Data

In this section, we present several experimental results demonstrating the effectiveness of the kernel dictionary-based methods for screen touch gesture recognition. In particular, we present results on the dataset described in the previous section, the Touchalytics dataset [5] and the BTAS 2013 dataset [4].

2.3.1 Experiment Setup

In this part, we give a detailed description of the experimental setup by specifying evaluation metrics, feature extraction, implementation details and different comparison strategies.

For a fair comparison, with all the datasets available, we extracted the same features on all the datasets, fixed the implementation details, optimized the parameters of every algorithm using cross validation, repeated the experiment multiple times by randomly splitting the data into training data and testing data and report the mean and standard deviation of the evaluation metrics.

Evaluation Metrics

Average Equal Error Rate (EER) and average F1 score are used to evaluate the performance of different methods. The EER is the error rate at which the probability of false acceptance rate is equal to the probability of false rejection rate. The lower the EER value, the higher the accuracy of the biometric system. The F1 score is defined as a harmonic mean of precision P and recall R

$$\text{F1 score} = \frac{2PR}{P+R},$$

where the precision P is the number of correct results divided by the number of all returned results and the recall R is the number of correct results divided by the number of results that should have been returned. The F1 score is always between 0 and 1. The higher the F1 score, better is the accuracy of the biometric system.

Implementation Details

The state of the art performance of touch gesture recognition achieved by kernel SVM with optimized parameters are shown in [5]and [16]. We compare the kernel SVM classifier with the kernel SRC (KSRC) classifier and the kernel dictionary-based classifier (KDTGR) for screen touch gestures.

We designed two types of experimental setups. For the first type of experiments on the datasets, we combined data from different tasks and sessions. and then we randomly split data for training and testing. As we were also interested in investigating how the environmental changes can affect the users' screen touch

behavior, for the second type of experiments on the datasets, we performed cross session recognition experiments where that training and testing samples are from different sessions. During testing, each user has his or her own test samples for genuine test and all the other users' test samples for impostor test which means a much larger number of samples were used for impostor test. In all the experiments, we used the histogram intersection kernel for KSRC and the Gaussian kernel with the optimized parameter was used for the kernel SVM. All the experiments were repeated 11 times.

Single-swipe vs. Multiple-swipe Classification

The performances of recognition algorithms is influenced by the number of swipes combined to predict a class label. For K-swipe classification, we first perform a single-swipe classification for all the K swipes and obtain the corresponding K predicted labels. Then by voting, we choose the one that appears most frequently as the final predicted class label. Here, we let K to be an odd integer. As K becomes larger, all the algorithms achieve better performance than the methods based on single-swipe classification. However, a large K implies longer time to collect swipes and predict the current class label. This is a tradeoff that one has to consider when designing an authentication system based on screen touch gestures.

2.3.2 Results on Touch Data Component Of UMDAA Dataset

For the first set of experiments using the new dataset, we randomly selected 80 swipes from each user to form the training matrix and used the remaining data for testing. Table 2.1 summarizes the results obtained by different methods for this experiment. For the single-swipe classification (row one in Table 2.1), rbfSVM performs the best. However, the average EER is very high for all the methods. This implies that authentication based simply on one swipe is very unreliable. As the number of swipes increase, KDTGR performs the best. This makes sense because by mapping the data onto a high-dimensional feature space and finding a compact representation by learning a dictionary in the feature space, one is able to find the common internal structure of the screen touch data. Classification based on the reconstruction error in the feature space is essentially improving the overall classification accuracy. As kernel SRC does not use dictionary learning step, it does not provide the best results for this dataset.

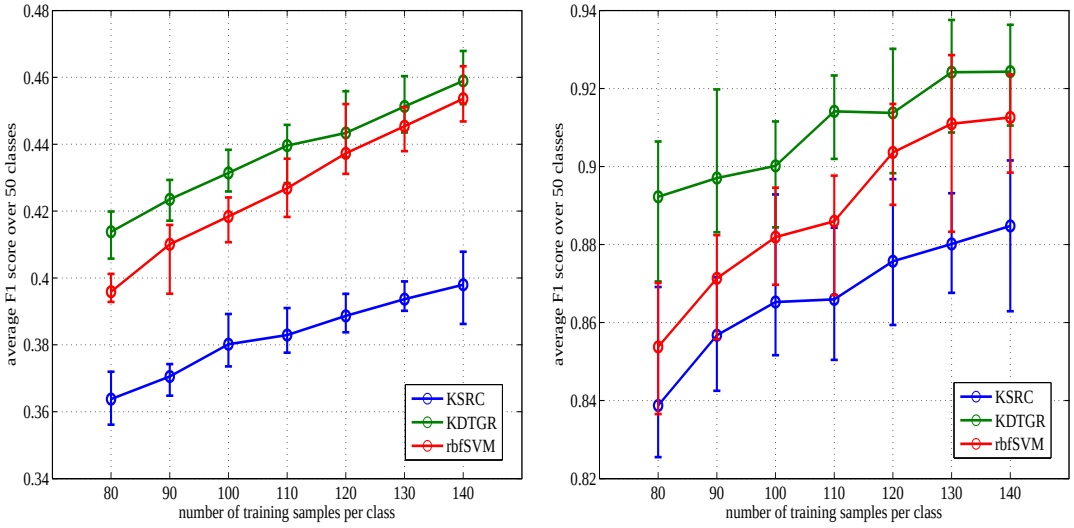
In the second set of experiments with the new dataset, we studied the performance of different classification methods as we increase the number of training samples. The average F1 score values for different number of training samples corresponding to a single-swipe and eleven-swipe classification are shown in Figure 2.1(a) and (b), respectively. As can be seen from these figures, KDTGR performs the best for both single-swipe and eleven-swipe classification. Furthermore, the average F1 score value increases as we increase the number of training samples for all the three classification methods. In particular, the average F1 score approaches 0.924, 0.913,

Swipes	KSRC	KDTGR	rbfSVM
1	29.86 ± 0.37	28.03 ± 0.22	17.41 ± 0.13
3	15.82 ± 0.30	12.92 ± 0.34	14.00 ± 0.27
5	9.71 ± 0.33	7.53 ± 0.31	8.56 ± 0.25
7	7.50 ± 0.32	5.59 ± 0.20	6.15 ± 0.27
9	5.85 ± 0.41	4.12 ± 0.22	4.75 ± 0.29
11	4.55 ± 0.32	2.91 ± 0.21	3.58 ± 0.26
13	3.40 ± 0.32	2.16 ± 0.14	2.66 ± 0.28
15	2.55 ± 0.40	1.43 ± 0.23	2.11 ± 0.27
17	1.98 ± 0.20	1.05 ± 0.20	1.43 ± 0.24
19	1.54 ± 0.25	0.77 ± 0.21	1.13 ± 0.22

Table 2.1: Average EER values (in %) for different classification methods on the new dataset.

0.885 for the KDTGR method, rbfSVM and the KSRC method, respectively when 140 samples are used for training for eleven-swipe classification.

Finally, in the last set of experiment with the new dataset, we performed cross-session experiments. In particular, since the new dataset contains data from three different sessions with different environmental conditions, we trained classifiers using the data from one session and test it on data from other session. We repeated this procedure for all the six different combinations of three sessions. For these experiments, we omitted eight users who have less than 70 swipes in any one of the three sessions. Then, we randomly selected 70 swipes for each user in one session to form the training data and randomly selected 70 swipes for each user in another session to form the test data. The average EER values for different cases for eleven-



(a) (b)

Figure 2.1: Average F1 score values (in %) for different classification methods on the new dataset as the number of training samples are increased. (a) Single-swipe classification. (b) Eleven-swipe classification.The figure is best viewed in color.

Case	KSRC	KDTGR	rbfSVM
$1 \rightarrow 2$	12.05 $\pm$1.21	9.90 $\pm$0.61	11.04 $\pm$1.13
$1 \rightarrow 3$	14.21 $\pm$1.10	11.72 $\pm$0.64	13.08 $\pm$1.12
$2 \rightarrow 1$	14.42 $\pm$0.79	11.69 $\pm$1.12	11.65 $\pm$1.07
$2 \rightarrow 3$	7.23 $\pm$0.53	5.64 $\pm$0.63	5.85 $\pm$0.66
$3 \rightarrow 1$	13.94 $\pm$1.60	11.60 $\pm$0.91	11.75 $\pm$0.97
$3 \rightarrow 2$	7.43 $\pm$0.87	4.88 $\pm$0.74	5.29 $\pm$0.75
1 2 3$\rightarrow$1 2 3	4.21 $\pm$0.67	2.62 $\pm$0.65	3.10 $\pm$0.30

Table 2.2: Average EER values (in %) for the cross-session experiments with the new dataset. In the first column of this table, $a \rightarrow b$ means that data from session a are used for training and data from session b are used for testing.

swipe cross-session classification experiments are summarized in Table 2.2. The last row of the Table shows the result which were obtained by combining data samples from different sessions together and then splitting them into training and testing data.

As can be seen from Table 2.2, on average the KDTGR method performed the best. When samples from all three sessions are used for training, all three classification methods performed well. This can be seen from the last row of the table. However, when classifiers are trained on data from one session and tested on the data from another session, the performance of all the three methods degraded noticeable.

Swipes	KSRC	KDTGR	rbfSVM
1	17.62 ± 0.45	17.69 ± 0.26	8.51 ± 0.13
3	6.13 ± 0.19	4.05 ± 0.097	4.16 ± 0.12
5	3.42 ± 0.13	2.29 ± 0.074	2.33 ± 0.11
7	2.19 ± 0.14	1.14 ± 0.13	1.25 ± 0.11
9	1.31 ± 0.099	0.60 ± 0.079	0.67 ± 0.088
11	0.85 ± 0.11	0.34 ± 0.084	0.36 ± 0.090
13	0.50 ± 0.082	0.16 ± 0.079	0.21 ± 0.074
15	0.35 ± 0.10	0.10 ± 0.062	0.16 ± 0.063
17	0.28 ± 0.082	0.051 ± 0.035	0.086 ± 0.043
19	0.18 ± 0.054	0.026 ± 0.025	0.060 ± 0.036

Table 2.3: Average EER values (in %) for different classification methods on the Touchalytics dataset.

2.3.3 Results On Touchalytics Dataset

Touchalytics dataset [5] consists of 41 users' touch data collected in two sessions separated by one week as described in the original paper. For each user, we randomly selected 80 swipes as training data and the remaining swipes as test data. Results are summarized in Table 2.3. As can be seen from this Table, on average, the KDTGR method performed the best. For a single-swipe classification, rbfSVM performed better than KSRC and KDTGR. As the number of swipes is increased, KDTGR outperformed the other methods.

In the Touchalytics dataset, Session 2 contains touch data from only 14 users. As a result, we did not perform the cross-session experiments on this dataset.

2.3.4 Results On BTAS 2013 Dataset

The BTAS 2013 dataset [4] is a large dataset which consists of data in two parts: 138 users' mobile touch data in portrait mode over 2 sessions and 59 users' mobile touch data in landscape mode over 2 different sessions.

Portrait Mode Cross-Session Experiment

Only one user had data with less than 80 swipes in any one of the 2 sessions. We omitted this user for the cross-session experiments. We randomly selected 80 swipes for each user in one session to form the training data and randomly selected 80 swipes for each user in the other session to form the test data. For comparison, we also considered the case where 80 training data and 80 testing data for each user were selected from both sessions. Table 2.4 shows the average EER values for different cases when 11 swipes were combined to make the final decision (e.g. eleven-swipe classification).

It is interesting to see that when data from both sessions are used, the EER values are the lowest. Similar to the observation we made in the experiments with the new dataset, as we train on the data from one session and test on the data from the other session, the performance of all the three classification methods degraded significantly.

Case	KSRC	KDTGR	rbfSVM
$1 \rightarrow 2$	23.51 ± 0.70	19.78 ± 0.65	20.67 ± 0.53
$2 \rightarrow 1$	23.83 ± 0.49	19.20 ± 0.72	20.06 ± 0.62
$1\ 2 \rightarrow 1\ 2$	8.94 ± 0.62	5.00 ± 0.46	5.86 ± 0.45

Table 2.4: Average EER values (in %) for the portrait mode cross-session experiments with the BTAS 2013 dataset. In the first column of this table, $a \rightarrow b$ means that data from session a are used for training and data from session b are used for testing.

Case	KSRC	KDTGR	rbfSVM
$1 \rightarrow 2$	14.25 ± 0.70	11.09 ± 0.98	13.19 ± 0.81
$2 \rightarrow 1$	13.70 ± 0.49	11.29 ± 0.54	12.04 ± 0.83
$1\ 2 \rightarrow 1\ 2$	4.06 ± 0.68	1.73 ± 0.44	2.18 ± 0.35

Table 2.5: Average EER values (in %) for the landscape mode cross-session experiments with the BTAS 2013 dataset. In the first column of this table, $a \rightarrow b$ means that data from session a are used for training and data from session b are used for testing.

Landscape Mode Cross-Session Experiment

Like before, we omitted six users who had fewer than 80 swipes in any one of the two sessions. We applied the same experiment setup as we did for the touch data in the portrait mode. Table 2.5 shows the average EER values for different cases when eleven swipes were combined to make the final decision. Again, the KDTGR method outperformed the other methods on this dataset.

2.4 Conclusion

In this chapter, we discussed the active authentication problem and proposed kernel sparse representation and kernel dictionary learning-based methods for touch gesture-based active user authentication. Experiments on screen touch data of UMDAA datasets as well as two publicly available screen touch datasets showed that the proposed kernel dictionary-based method performed favorably over other compared methods. Cross-session experiments showed that there is a significant drop in the performance of all the methods. This problem can be viewed as *domain adaptation* [26] or *dataset bias* problem [27] which have been studied in machine learning, natural language processing and computer vision. The following chapter will address this problem.

Chapter 3: Domain Adaptive Sparse Representation-Based Classification

3.1 Introduction

In biometrics recognition, one is often faced with scenarios where the training data used to learn a recognition engine has a different distribution from the test data. Examples of such cases include: recognizing and detecting faces under different lighting conditions and poses while the algorithms are trained on well-illuminated frontal faces, recognizing low-resolution face images when recognition algorithms are instead optimized for high-resolution images, recognizing and detecting human faces on infrared images while the algorithms are optimized for color images, etc. Regardless of the specific cause, any distribution change that occurs after learning a classifier can degrade its performance at test time. Domain adaptation essentially tries to mitigate this dataset shift problem.

We propose an approach to the problem of domain adaptation based on sparse representation. Our method learns projections of data in different domains in a way that preserves the sparse structure of data in the low-dimensional space. We develop an efficient optimization method based on Alternating Direction Method

of Multipliers (ADMM) and the Method of Splitting Orthogonality Constraints (SOC) for solving the resulting optimization problem. One of the advantages of the proposed method compared to other dictionary-based domain adaptation methods is that it is very efficient as it does not require learning a dictionary. Our method can be viewed as a generalization of the Sparse Representation-based Classification (SRC) [9] that accounts for domain shift. An overview of the proposed method is shown in Figure 3.1.

The rest of the chapter is organized as follows: In Section 3.2, we review some recent domain adaptation methods. The proposed domain adaptive sparse representation-based classification problem is formulated in Section 3.3. Details of the optimization algorithm are given in Section 3.4. Experimental results are given in Section 3.5. Finally, Section 3.6 concludes the this chapter with a brief summary and discussion.

3.2 Related Work

Various domain adaptation methods have been proposed in the computer vision and machine learning literature. One of the simplest domain adaptation approaches is the feature augmentation work proposed in [28]. The goal is to make a domain specific copy of the original features for each domain. This work was extended for the heterogeneous data in [29]. The idea of feature augmentation has also been extended to consider a manifold of intermediate domains [30]. Rather than working with information conveyed by the source and target domains alone, [30] pro-

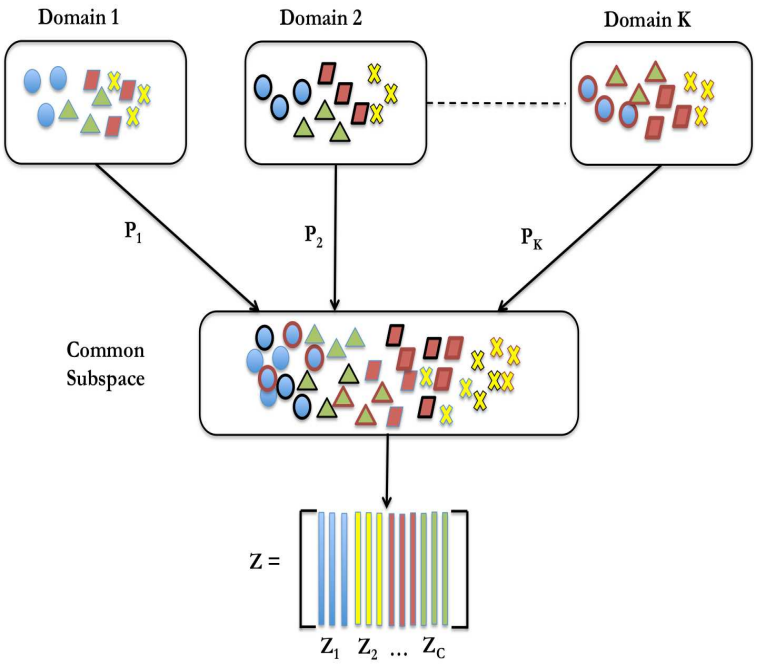


Figure 3.1: An overview of learning domain adaptive sparse representations.

posed an incremental learning technique based on gradually following the geodesic path between the source and target domains. Geodesic flows were used to derive intermediate subspaces that interpolate between the source and target domains. Recently, the approach of [30] was kernelized and extended to the infinite case, defining a new kernel equivalent to integrating over all common subspaces that lie on the geodesic flow connecting the source and target subspaces, respectively [31].

Various feature transformation-based approaches have also been proposed in the literature [32], [33], [34]. The idea behind this method is to adapt features across general image domains by learning transformations. Another class of domain adaptation algorithms is based on parameter adaptation in which the Support Vector Machine (SVM) type of algorithms are proposed for domain adaptation. Algorithms

such as adaptive SVM [35], domain transfer SVM [36], max-margin domain transfer [37] and domain adaptive multiple kernel learning [38] fall under this category.

Dictionary learning-based methods have also gained a lot of attention in recent years for domain adaptation. In [39], the idea of sparse domain transfer under the framework of dictionary learning was proposed for image super-resolution and photo-sketch synthesis. A technique for jointly learning transformations of data in source and target domains, and a latent discriminative dictionary that can succinctly represent both domains in the projected low-dimensional space was proposed in [40]. In [41], a function learning framework was presented for the task of transforming a dictionary learned from one visual domain to the other, while maintaining a domain-invariant representation of a signal. Another approach [42] proposed using concepts from dictionary learning to generate intermediate domains that bridge the domain shift. See [43], [44] and [45] for more detailed discussion of recent domain adaptation approaches.

3.3 Problem Formulation

In this section, we formulate the domain adaptation problems. For simplicity, we begin with two domains adaptation problems by specifying all the details. This is followed by a straightforward generalization to multiple domains.

3.3.1 Two Domains Formulation

Let $\{(\mathbf{y}_i^{d_1}, c_i^{d_1})\}_{i=1}^{N_1}$, denote the collection of N_1 labeled data from the domain $\mathcal{D}_1$. Here, $\mathbf{y}_i^{d_1} \in \mathbb{R}^{M_1}$ is referred to as the ith observation and $c_i^{d_1}$ is the corresponding class label. Labeled data from the domain $\mathcal{D}_2$ is denoted by $\{(\mathbf{y}_i^{d_2}, c_i^{d_2})\}_{i=1}^{N_2}$ where $\mathbf{y}_i^{d_2} \in \mathbb{R}^{M_2}$. Denote

$$\mathbf{Y}_1 = [\mathbf{y}_1^{d_1}, \cdots, \mathbf{y}_{N_1}^{d_1}] \in \mathbb{R}^{M_1 \times N_1}$$

as the matrix of N_1 data points from $\mathcal{D}_1$. Similarly, denote

$$\mathbf{Y}_2 = [\mathbf{y}_1^{d_2}, \cdots, \mathbf{y}_{N_2}^{d_2}] \in \mathbb{R}^{M_2 \times N_2}$$

as the matrix of N_2 data from $\mathcal{D}_2$. It is assumed that the data from both domains pertain to C subjects or classes. We assume that there is always a relatively large amount of labeled data in the source domain and a small amount of labeled data in the target domain. As a result, if $\mathcal{D}_1$ corresponds to the source domain and $\mathcal{D}_2$ corresponds to the target domain then $N_1 \gg N_2$.

Let $\mathbf{P}_1 \in \mathbb{R}^{m \times M_1}$ and $\mathbf{P}_2 \in \mathbb{R}^{m \times M_2}$ be the mappings represented as matrices that project the data from $\mathcal{D}_1$ and $\mathcal{D}_2$ to a common m-dimensional space, respectively. As a results, $\mathbf{P}_1\mathbf{Y}_1$ and $\mathbf{P}_2\mathbf{Y}_2$ lie on an m-dimensional space. Let

$$\mathbf{Z} = [\mathbf{P}_1\mathbf{Y}_1, \mathbf{P}_2\mathbf{Y}_2] = [\mathbf{z}_1, \cdots, \mathbf{z}_{N_1+N_2}] \in \mathbb{R}^{m \times (N_1+N_2)}$$

denote the samples in the m-dimensional space. In our method, we want to take advantage of the *self-expressiveness property* of the data in the low-dimensional space [46]. That is, each data $\mathbf{z}_i$ can be efficiently reconstructed by a combination

of other points in $\mathbf{Z}$. More precisely, $\mathbf{z}_i$ can be written as

$$\mathbf{z}_i = \mathbf{Z}\mathbf{b}_i, \quad b_{i,i} = 0 \tag{3.1}$$

where $\mathbf{b}_i = [b_{i,1}, b_{i,2}, \cdots, b_{i,N_1+N_2}]^T$. Here, the constraint $b_{i,i} = 0$ eliminates the trivial solution that arises as a result of representing a point as a linear combination of itself in the projected m-dimensional space. The assumption that $N_1 + N_2 \gg m$ results in many solutions for (3.1). One can look for the sparsest solution and restrict the set of solutions by minimizing the following sparse optimization problem

$$\begin{aligned} &\min \|\mathbf{b}_i\|_1 \\ &s.t.\ \mathbf{z}_i = \mathbf{Z}\mathbf{b}_i, \quad b_{i,i} = 0 \end{aligned} \tag{3.2}$$

where $\|\mathbf{b}_i\|_1 = \sum_j |b_{i,j}|$ is the ℓ_1-norm of $\mathbf{b}_i$. This problem can be solved using convex optimization methods. One can rewrite the sparse optimization problem (3.2) for all samples in the m-dimensional space as

$$\begin{aligned} &\min \|\mathbf{B}\|_1 \\ &s.t.\ \mathbf{Z} = \mathbf{Z}\mathbf{B} \\ &\quad \operatorname{diag}(\mathbf{B}) = \mathbf{0} \end{aligned} \tag{3.3}$$

where $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{N_1+N_2}] \in \mathbb{R}^{(N_1+N_2)\times(N_1+N_2)}$ is the coefficient matrix whose ith column is the sparse coefficient corresponding to $\mathbf{z}_i$ and $\operatorname{diag}(\mathbf{B})$ is the vector of the diagonal elements of $\mathbf{B}$.

In our approach, we desire to learn projections $\mathbf{P}_1$ and $\mathbf{P}_2$ along with the sparse

coefficient matrix $\mathbf{B}$ simultaneously by minimizing the following cost function

$$\begin{aligned} \min_{\mathbf{P},\mathbf{B}} \ & \mathcal{C}_1(\mathbf{P},\mathbf{Y},\mathbf{B}) + \beta\mathcal{C}_2(\mathbf{P},\mathbf{Y}) + \mu\|\mathbf{PY}\|_F^2 + \lambda\|\mathbf{B}\|_1 \\ s.t. \ & \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I} \\ & \text{diag}(\mathbf{B}) = \mathbf{0} \end{aligned} \tag{3.4}$$

where β, μ and λ are the regularization parameters, $\mathcal{C}_1(\mathbf{P},\mathbf{Y},\mathbf{B}) = \|\mathbf{PY} - \mathbf{PYB}\|_F^2$ and $\mathcal{C}_2(\mathbf{P},\mathbf{Y}) = \|\mathbf{Y}_1 - \mathbf{P}_1^T\mathbf{P}_1\mathbf{Y}_1\|_F^2 + \|\mathbf{Y}_2 - \mathbf{P}_2^T\mathbf{P}_2\mathbf{Y}_2\|_F^2$. After ignoring the constant terms in $\mathbf{Y}$, $\mathcal{C}_2$ can be rewritten as

$$\mathcal{C}_2(\mathbf{P},\mathbf{Y}) = -\text{tr}((\mathbf{PY})(\mathbf{PY})^T).$$

Here $\mathbf{P}$ and $\mathbf{Y}$ are defined as

$$\mathbf{P} = [\mathbf{P}_1 \ \ \mathbf{P}_2] \in \mathbb{R}^{m\times(M_1+M_2)}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_2 \end{bmatrix} \in \mathbb{R}^{(M_1+M_2)\times(N_1+N_2)}.$$

The first part of the cost function $\mathcal{C}_1$ with the constraint that $\text{diag}(\mathbf{B}) = \mathbf{0}$ essentially exploits the self-expressiveness property of the data in the sense that each data point can be efficiently reconstructed by a combination of other points in the database. Similar ideas have been explored for subspace clustering using sparse representations in [46]. The second term $\mathcal{C}_2$ is a PCA-like regularization term, ensures that the projection does not loose too much information available in the original domain. Finally, $\|\mathbf{PY}\|_F^2$ is added to ensure the convexity of the cost function.

3.3.2 Multi-Domain Formulation

The above formulation can be extended from two domains to multiple domains. For K domain problem, we have data $\mathbf{Y}_1, \cdots, \mathbf{Y}_K$ from K different domains $\mathcal{D}_1, \cdots, \mathcal{D}_K$ and one can simply construct $\mathbf{P}$ and $\mathbf{Y}$ as

$$\mathbf{P} = [\mathbf{P}_1 \cdots \mathbf{P}_K], \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{Y}_K \end{bmatrix}.$$

With these definitions, (3.4) can be extended to multiple domains. Note that we do not require the dimensions from different domains to be the same. As a result, our method can be viewed as a heterogeneous domain adaptation method [45].

3.4 Optimization

We solve the optimization problem (3.4) by optimizing over $\mathbf{P}$ and $\mathbf{B}$ iteratively. Note that the optimization problem is non-convex. However, numerical simulations have shown that the algorithm usually converges to a local minimum in a few iterations.

3.4.1 Update $\mathbf{B}$

In this step, we assume that $\mathbf{P}$ is fixed. As a result, the following problem needs to be solved

$$\min_{\mathbf{B}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \lambda \|\mathbf{B}\|_1 \tag{3.5}$$
$$s.t.\ \mathrm{diag}(\mathbf{B}) = \mathbf{0}.$$

This problem is similar to the Sparse Subspace Clustering (SSC) problem [46] which can be efficiently solved using the ADMM method [47].

3.4.2 Update $\mathbf{P}$

For a fixed $\mathbf{B}$, we have to solve the following problem to obtain $\mathbf{P}$

$$\min_{\mathbf{P}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \beta \mathcal{C}_2(\mathbf{P}, \mathbf{Y}) + \mu \|\mathbf{PY}\|_F^2 \tag{3.6}$$
$$s.t.\ \mathbf{P}_1 \mathbf{P}_1^T = \mathbf{P}_2 \mathbf{P}_2^T = \mathbf{I}.$$

The cost function of (3.6) can be rewritten as

$$\begin{aligned}
&\mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \beta \mathcal{C}_2(\mathbf{P}, \mathbf{Y}) + \mu \|\mathbf{PY}\|_F^2 \\
&= \|\mathbf{PY} - \mathbf{PYB}\|_F^2 + (\mu - \beta)\mathrm{tr}((\mathbf{PY})(\mathbf{PY})^T) \\
&= \mathrm{tr}[(\mathbf{PY} - \mathbf{PYB})^T(\mathbf{PY} - \mathbf{PYB}) + (\mu - \beta)(\mathbf{PY})(\mathbf{PY})^T] \\
&= \mathrm{tr}[\mathbf{P}(\mathbf{Y}(\mathbf{I} - 2\mathbf{B} + \mathbf{BB}^T + (\mu - \beta)\mathbf{I})\mathbf{Y}^T)\mathbf{P}^T].
\end{aligned}$$

Let $\mathbf{H} = \mathbf{Y}(\mathbf{I} - 2\mathbf{B} + \mathbf{B}\mathbf{B}^T + (\mu - \beta)\mathbf{I})\mathbf{Y}^T$ be $\sum_i M_i \times \sum_i M_i$ matrix. Then, the optimization problem (3.6) can be rewritten as

$$\min_{\mathbf{P}} \operatorname{tr}[\mathbf{P}\mathbf{H}\mathbf{P}^T] \tag{3.7}$$
$$s.t.\ \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I}.$$

This optimization problem involves trace minimization with multiple orthogonality constraints. The cost function is convex when $\mathbf{H}$ is positive semi-definite; however multiple orthogonality constraints make the problem not convex and we cannot directly solve it as a classical eigen problem.

In what follows, we present the method of SOC [48] for solving this problem . Let $\mathbf{O} = \mathbf{P}^T$. Then, the trace minimization problem (3.7) with K orthogonality constants can be rewritten as

$$\min_{\mathbf{O}} g(\mathbf{O}_1, \cdots, \mathbf{O}_K; \mathbf{H}) \tag{3.8}$$
$$s.t.\ \mathbf{O}_i^T\mathbf{O}_i = \mathbf{I}\ \ \forall i = 1, \cdots, K,$$

where $\mathbf{O}_i \in \mathbb{R}^{M_i \times m}$, $m \leq \min\{M_1, M_2, \cdots, M_K\}$,

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} & \cdots & \mathbf{H}_{1K} \\ \mathbf{H}_{21} & \mathbf{H}_{22} & \cdots & \mathbf{H}_{2K} \\ \mathbf{H}_{K1} & \mathbf{H}_{K2} & \cdots & \mathbf{H}_{KK} \end{bmatrix} \in \mathbb{R}^{\sum_i M_i \times \sum_i M_i},$$

$\mathbf{H}_{ij} \in \mathbb{R}^{M_i \times M_j}$ and $g(\mathbf{O}_1, \cdots, \mathbf{O}_K; \mathbf{H}) = \operatorname{tr}[\mathbf{O}^T\mathbf{H}\mathbf{O}]$. The SOC method solves the orthogonality constrained problems by iteratively optimizing the unconstrained and quadratic problems with analytic solutions using the combination of variable splitting and Bregman iterations [49]. It consists of three main steps.

Update $\mathbf{O}_i$: For updating $\mathbf{O}_i$ one at a time, we need to solve the following sub optimization problem

$$\mathbf{O}_i^t = \arg\min_{\mathbf{O}_i} g(\mathbf{O}_1^{t-1}, \cdots, \mathbf{O}_K^{t-1}) + \frac{\gamma}{2}\|\mathbf{O}_i - \mathbf{Q}_i^{t-1} + \mathbf{R}_i^{t-1}\|_F^2.$$

Where γ is a positive parameter that can be tuned. By taking the first derivative and setting it equal to zero, we get

$$\mathbf{O}_i^t = \left(\frac{\gamma}{2}\mathbf{I} + \mathbf{H}_{ii}\right)^{-1} \left[\frac{\gamma}{2}(\mathbf{Q}_i^{t-1} - \mathbf{R}_i^{t-1}) - \sum_{\substack{j=1 \\ j\neq i}}^{K} \mathbf{H}_{ij}\mathbf{O}_j^{t-1}\right].$$

Update $\mathbf{Q}_i$: In order to update $\mathbf{Q}_i$, we need to solve the following optimization problem

$$\mathbf{Q}_i^t = \arg\min_{\mathbf{Q}} \frac{\gamma}{2}\|\mathbf{Q}_i - (\mathbf{O}_i^t - \mathbf{R}_i^{t-1})\|_F^2 \tag{3.9}$$

$$s.t\ \mathbf{Q}_i^T\mathbf{Q}_i = \mathbf{I}$$

whose closed form solution is obtained as

$$\mathbf{Q}_i^t = \mathbf{U}_i\mathbf{I}_{M_i\times m}\mathbf{V}_i^T,$$

where $\mathbf{U}_i\mathbf{D}_i\mathbf{V}_i^T$ is the Singular Value Decomposition (SVD) of $(\mathbf{O_i^t} - \mathbf{R_i^{t-1}})$ and $\mathbf{U}_i \in \mathbb{R}^{M_i\times M_i}, \mathbf{D}_i \in \mathbb{R}^{M_i\times m}$, $\mathbf{V}_i \in \mathbb{R}^{m\times m}$.

Update $\mathbf{R}_i$: Finally, having updated $\mathbf{Q}_i$ and $\mathbf{O}_i$, $\mathbf{R}_i$ is updated as follows

$$\mathbf{R}_i^t = \mathbf{R}_i^{t-1} + \left(\mathbf{O}_i^t - \mathbf{Q}_i^t\right).$$

The entire procedure for solving (3.8) using the method of SOC is summarized in Algorithm 1.

The Domain Adaptive Sparse Representation learning process is summarized in Algorithm 2.

Algorithm 1: The method of SOC for solving (3.8).

Input: $\mathbf{O}, \mathbf{H}, \gamma$

Initialization: $\mathbf{R}_0, \mathbf{O}_0, \mathbf{Q}_0$

While not converge do

1. Update $\mathbf{O}_i$:

$$\mathbf{O}_i^t = \left(\frac{\gamma}{2}\mathbf{I} + \mathbf{H}_{ii}\right)^{-1} \left[\frac{\gamma}{2}(\mathbf{Q}_i^{t-1} - \mathbf{R}_i^{t-1}) - \sum_{\substack{j=1 \\ j\neq i}}^{K} \mathbf{H}_{ij}\mathbf{O}_j^{t-1}\right]$$

2. Update $\mathbf{Q}_i$:

$$\mathbf{Q}_i^t = \mathbf{U}_i \mathbf{I}_{M_i \times m} \mathbf{V}_i^T$$

3. Update $\mathbf{R}_i$:

$$\mathbf{R}_i^t = \mathbf{R}_i^{t-1} + \left(\mathbf{O}_i^t - \mathbf{Q}_i^t\right)$$

Output: $\hat{\mathbf{O}} = [\mathbf{O}_1^t, \cdots, \mathbf{O}_K^t]$

Algorithm 2: Learning Domain Adaptive Sparse Representation

Input: Data $\mathbf{Y}_1, \cdots, \mathbf{Y}_K$ and corresponding class labels, β, μ, λ

Initialization: $\mathbf{P}$

Until convergence do

1. Update $\mathbf{B}$: Solve the following ℓ_1 minimization problem using the ADMM procedure described in [46]

$$\min_{\mathbf{B}} \mathcal{C}_1(\mathbf{P}, \mathbf{Y}, \mathbf{B}) + \lambda\|\mathbf{B}\|_1 \;\; s.t. \;\; \mathrm{diag}(\mathbf{B}) = \mathbf{0}$$

2. Update $\mathbf{P}$: Solve the following optimization problem using the method of SOC as summarized in Algorithm 1.

$$\min_{\mathbf{P}} \mathrm{tr}[\mathbf{P}\mathbf{H}\mathbf{P}^T] \;\; s.t. \;\; \mathbf{P}_1\mathbf{P}_1^T = \mathbf{P}_2\mathbf{P}_2^T = \mathbf{I}$$

Output: $\hat{\mathbf{B}}$ and $\hat{\mathbf{P}} = [\hat{\mathbf{P}}_1, \hat{\mathbf{P}}_2, \cdots, \hat{\mathbf{P}}_K]$

3.4.3 Domain Adaptive Sparse Representation-Based Classification

Given a test sample $\mathbf{y}_t$ from domain k, we propose the following steps for classification.

1. Compute the embeddings of all the training samples from different domains in the common m-dimensional subspace using the corresponding projections as $\mathbf{P}_i\mathbf{Y}_i \in m \times N_i$.

2. Using the label information, form a training matrix in the low-dimensional subspace as follows

$$\mathbf{Z} = [\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_C] \in \mathbb{R}^{m \times \sum_i N_i},$$

where $\mathbf{Z}_i$ is the matrix corresponding to the training samples from class i in the m-dimensional space.

3. Compute the embedding of the test sample $\mathbf{y}_t$ in the common m-dimensional subspace using the projection $\mathbf{P}_k$ as

$$\mathbf{z}_t = \mathbf{P}_k\mathbf{y}_t.$$

4. Compute the sparse coefficient $\hat{\boldsymbol{\alpha}}_t$ of the embedded sample $\mathbf{z}_t$ over dictionary $\mathbf{Z}$ by solving the following optimization problem

$$\hat{\boldsymbol{\alpha}}_t = \min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}_t|_0 \;\; s.t. \;\; \|\mathbf{z}_t - \mathbf{Z}\boldsymbol{\alpha}_t\|_F^2 \leq \eta, \tag{3.10}$$

where η is the noise level and $\|\mathbf{x}\|_0$ is the ℓ_0-norm of $\mathbf{x}$ which counts the number of non-zero elements in $\mathbf{x}$. We use the Orthogonal Matching Pursuit (OMP) algorithm [22] to solve (3.10).

5. The sample can be assigned to class i if the reconstruction using the samples corresponding to class i is minimum

$$\text{Output label} = \hat{i} = \arg\min_i \|\mathbf{z}_t - \mathbf{Z}\delta_i(\hat{\boldsymbol{\alpha}}_t)\|_F^2,$$

where $\delta_i(\cdot)$ is the characteristic function that selects the coefficients associated with the ith class.

3.5 Experimental Results

In this section, we evaluate the proposed algorithm on the UMDAA dataset. For the domain adaptation experiments, we sampled a subset from UMDAA dataset: (1) for face component, we selected 30 faces from each session for each user. As a result, in total we selected 4500 face images for 50 users across three different domains. For the touch signature, we also selected 4500 touch swipes of 50 users across three domains. All the experiment done will be based on these selected 4500 face images and 4500 touch swipes.

Because the underlying characteristics of data collected in different sessions with different ambient conditions is very different, data in different sessions can be viewed as data from different domains. Therefore, it is appropriate to apply domain adaptation methods to design classifiers that are robust to different sessions (domains).

3.5.1 Experimental Setup

Algorithms and Implementation details

We compare our method with several recent domain adaptation algorithms including a metric learning-based method [32], a manifold-based method [30], and dictionary learning-based methods [42], [40]. We also use the SRC method [9] as a baseline for comparison. For SRC, data from different domains are used without domain adaptation. This method essentially shows the performance of a sparsity-based method when training and test samples come from different domains. Comparison of our Domain-Adaptive SRC (DASRC) method with SRC will validate the effectiveness of the proposed domain adaptation approach.

For the proposed DASRC algorithm, we choose $\mu - \beta = 4.5$, $\lambda = 50$ and $\gamma = 60$ which are the tuned results from the cross validation experiments. Parameters for other domain adaptation methods were optimized according to the discussion provided in the corresponding papers.

3.5.2 Single-source Domain Adaptation Experiments

Following the standard domain adaptation protocol, we selected 20 samples for each user from one session as the source domain and 5 samples for each user from another session as the target domain to form the training data. The remaining data from the target domain were used for testing. We randomly split the training and testing datasets, and repeated each experiment 10 times and report the mean and

Methods	1 → 2	1 → 3	2 → 3	2 → 1	3 → 1	3 → 2	Average
SRC [9]	73.52 ±1.49	85.12 ± 1.04	83.98± 0.91	80.83 ±1.08	80.73 ±1.35	72.57 ±1.13	79.46
Metric [32]	73.19 ±1.95	84.54 ± 1.27	80.36 ± 2.92	78.83 ±4.06	85.45 ±1.15	73.61 ±2.18	79.33
SGF [30]	56.57 ±1.22	62.58 ± 1.13	60.90 ± 1.05	54.94 ±2.19	65.66 ±1.75	62.69 ±1.33	60.56
SDDL [40]	55.48 ±4.40	71.67 ± 4.14	75.67 ± 3.72	71.71 ±4.46	77.74 ±4.15	66.74 ±2.91	69.84
Dict [42]	66.13 ±1.40	78.61 ± 1.42	76.26 ± 0.63	72.30 ±1.24	78.18 ±1.50	71.15 ±1.24	73.77
DASRC	**81.39 ± 1.66**	**89.06 ± 1.31**	**89.70 ± 1.05**	**87.36 ± 0.82**	**86.92 ± 0.99**	**82.16 ± 0.69**	**86.10**

Table 3.1: Recognition accuracy on target domain with semi-supervised adaptation for the face component.

the standard deviation of the classification accuracy. Since we have three sessions, there are six different combinations of source and target domains. The performance of our proposed method is compared with other domain adaptation methods for the face and the touch data in Table 3.1 and Table 3.2, respectively.

As can be seen from these tables, the proposed DASRC method outperforms the other methods on all six domain pairs. In some cases the improvement is over 10% over other methods. Furthermore, comparison with the SRC method shows that the sparse coding framework is insufficient when the test data has different characteristics than the data used for training. Also, the performance on faces is better than the performance on touch gestures.

3.5.3 Multi-source Domain Adaptation Experiments

For multi-source domain adaptation experiments, we selected 20 samples for each user from source domains and 5 samples for each user from the target domain

Methods	1 → 2	1 → 3	2 → 3	2 → 1	3 → 1	3 → 2	Average
SRC [9]	35.48 ±1.49	37.50 ± 0.86	40.18± 1.27	36.99 ±1.10	37.57 ±1.23	38.50 ±0.73	37.70
Metric [32]	24.58 ±1.75	25.71 ± 0.92	29.58 ± 2.22	22.45 ±2.07	24.25 ±1.90	28.59 ±1.47	25.86
SGF [30]	37.88 ±1.18	35.47 ± 1.25	37.00 ± 0.97	37.08 ±1.28	36.10 ±1.20	41.54 ±1.22	37.51
SDDL [40]	39.49 ±2.73	41.86 ± 2.36	42.28± 2.38	38.71±3.65	39.66 ±2.90	38.98 ±3.26	40.16
Dict [42]	30.31 ±1.39	31.00 ± 0.74	34.74 ± 1.05	30.58 ±0.94	32.55 ±0.73	36.21 ±0.82	32.57
DASRC	**41.54 ± 1.89**	**44.34 ± 1.66**	**44.77 ± 1.17**	**41.58 ± 1.35**	**41.82 ± 1.61**	**42.30 ± 1.50**	**42.74**

Table 3.2: Recognition accuracy on target domain with semi-supervised adaptation for the touch component.

Methods	1 2 → 3	1 3 → 2	2 3 → 1	Average
SRC [9]	89.68 ±0.83	81.14 ± 0.86	88.20 ± 0.76	86.34
SGF [30]	69.57 ±1.35	64.05 ± 1.21	62.21 ± 2.12	65.28
SDDL [40]	75.08 ±3.82	55.34 ± 2.34	72.86 ± 3.27	67.76
LMSDA [50]	82.48 ±1.04	70.17 ± 0.66	77.18 ± 1.18	76.61
DASRC	**90.94 ± 0.86**	**83.03 ± 0.74**	**88.44 ± 0.68**	**87.47**

Table 3.3: Multi-source domain adaptation on face data.

to form the training data. The remaining data from the target domain were used for testing. Like before, we repeated each experiment 10 times and report the mean and the standard deviation of the classification accuracy. Since we have three sessions, there are three different combinations of two source domains and one target domain. The experimental results comparing the proposed method with the other multi-source domain adaptation methods on the face data and the touch data are shown in Table 3.3 and Table 3.4, respectively.

Again, our DASRC method performs better than other methods on all possible

Methods	1 2 → 3	1 3 → 2	2 3 → 1	Average
SRC [9]	39.88 ±1.10	38.26 ± 0.63	36.68 ± 1.28	38.27
SGF [30]	39.04 ±0.99	39.13 ± 1.11	35.96 ± 0.94	38.04
SDDL [40]	34.66 ±1.50	31.21 ± 2.98	31.26 ± 2.56	32.38
LMSDA [50]	40.86 ±1.21	39.20 ± 0.79	37.42 ± 1.22	39.16
DASRC	**43.62 ± 1.75**	**42.17 ± 1.14**	**42.40 ± 0.83**	**42.73**

Table 3.4: Multi-source domain adaptation on touch data.

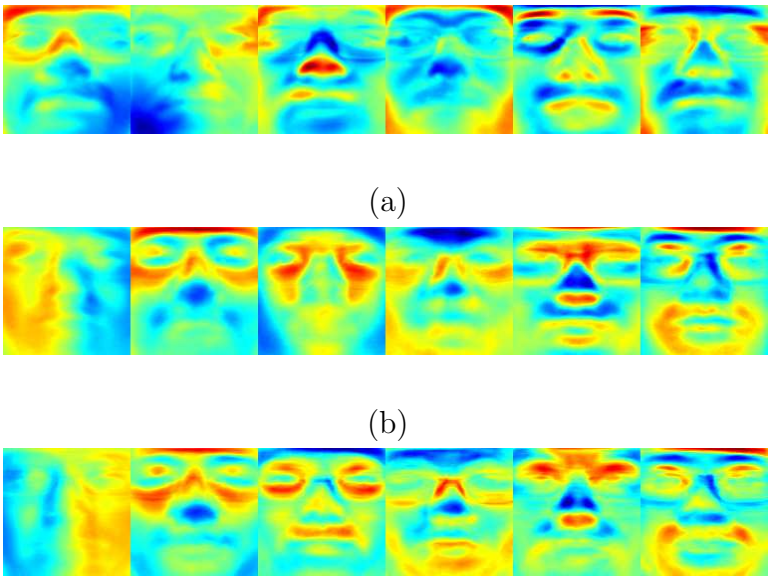
combinations. An interesting observation is that increasing the number of domains can be helpful, especially when compared to a single source and single target cases. This can be seen by comparing Tables 3.1 and 3.2 with tables 3.3 and 3.4. The gain is more apparent for faces.

3.5.4 Further Discussions And Analysis

Visualization of the Projection Matrices

To gain additional insights regarding our method, we investigated the projection matrices $\mathbf{P}_i \in \mathbb{R}^{m \times M_i}, \forall i = 1, \cdots, K$ learned by our method in the case of multi-source domain adaptation using faces. For better visualization, we used grayscale face images rescaled to 128×128 from the original preprocessed face images of size $192 \times 168 \times 3$. We followed the multi-source domain adaptation experiment setup as described above. We chose session 1 and 2 to be the source domains and session 3 to be the target domain. We first randomly selected 20 images per subject in each source domain, and five images per subject in the target domain, and then fed these

images to our proposed algorithm to learn the projection matrices $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$. Figure 3.2 shows the first six rows of the learned projection matrices reshaped as images. As can be seen from this figures, the projection matrices learn the internal structure of the different domains and can capture the shape, illumination and pose information. As a result, we are able to find better sparse representation in the projected m-dimensional space.

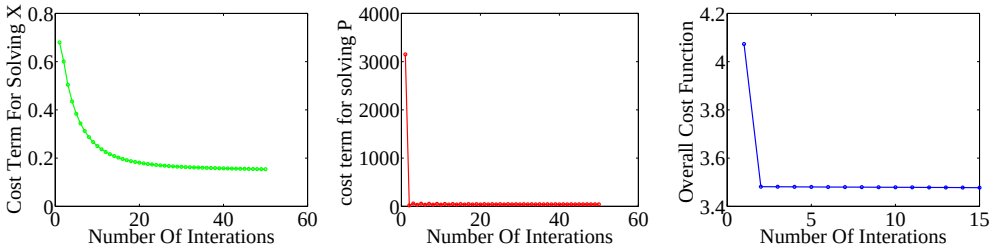


(a)

(b)

(c)

Figure 3.2: First six components of the learned projection matrices for the multi-source domain adaptation experiment. (a) Components from $\mathbf{P}_1$, (b) Components from $\mathbf{P}_2$. (c) Components from $\mathbf{P}_3$.



(a) (b) (c)

Figure 3.3: Objective function versus number of iterations of the proposed optimization problems. (a) The ADMM method for solving (3.5). (b) The method of SOC for solving the trace minimization problem with multiple orthogonality constraints (3.6). (c) The proposed problem (3.4).

Runtime Analysis and Computational Issue

In this section, we study the convergence properties of the proposed method and briefly discuss the computational complexity compared to the dictionary-based domain adaptation algorithms [42], [40].

As discussed earlier, our method is non-convex and often converges to a local minimum in a few iterations. To empirically show the convergence of our method, in Fig 3.3(a)-(c), we present the objective function vs iteration plots for the ADMM method for solving (3.5), the method of SOC for solving the trace minimization problem with multiple orthogonality constraints (3.6) and our proposed problem (3.4), respectively. As can be seen from this figure, both sub optimization problems as well as our overall algorithm do converge in a few iterations. Furthermore, compared to the previously proposed dictionary-based domain adaptation methods, our

method is very efficient. On average, the proposed method takes about $6.5ms$ to recognize a test image of size 24×21 compared to $26ms$ and $11ms$ for [42] and [40], respectively. Experiments were done in 64bit Matlab R2013a environment on a laptop with 2.9GHz Intel Core i7-3520M CPU and 8GB Memory.

3.6 Conclusion

In this chapter, a sparse representation-based classification algorithm was proposed for domain adaptation and an efficient iterative method based on ADMM and SOC methods was derived for solving the proposed optimization problem. This domain adaptation algorithm was evaluated on face component and screen touch component of the UMDAA Dataset. Experimental results showed that the proposed algorithm can help to alleviate the drop of the classification performance when training and test data comes from different domains (conditions) and it outperformed many state-of-the-art domain adaptation algorithms.

Chapter 4: Low-Rank and Joint Sparse Representations for Multimodal Recognition

4.1 Introduction

Developments in sensing and communication technologies have led to an explosion in the availability of data from multiple sources and modalities. Millions of sensors of different types have been installed in buildings, streets, and airports around the world that are capable of capturing multimodal information such as light, depth and heat. This has resulted in the development of various multi-sensor fusion methods [51], [52].

The idea of fusing multiple sources or modalities to achieve better performance compared to using a single modality alone is appealing. In particular, multimodal classification has received a lot of attention where one uses information from various modalities recording the same physical event to achieve improved classification performance. Many practical systems are multimodal systems. For example, in multimodal biometrics systems, similarity scores generated by multiple features extracted from face, fingerprints and iris are integrated for identity recognition [53], [54]. One advantage of multimodal biometrics systems is that they are less vulnerable to spoof

attacks.

In recent years, sparse and low-rank representations have been explored in problems such as matrix recovery [55], [56], [57], compressive sensing [58], regression [59], and subspace clustering [46], [60], [61], [62]. In particular, a low-rank and joint sparse representation-based method was proposed in [58] to recover hyperspectral images from very few number of noisy compressive measurements. A low-rank sparse subspace clustering (LRSSC) method was proposed in [62] that simultaneously enforces low-rank and sparse constraints on the representation matrix for subspace clustering. The trade-off between self-expressiveness property and graph-connectivity was analyzed and LRSSC was shown to take advantage of both low-rank and sparse constraints to yield improved clustering performance.

Motivated by recent developments and applications of low-rank and joint sparse representations [58], [62], [63], [64], we propose multimodal feature-level fusion methods by simultaneously enforcing low-rank and joint sparsity constraints across the representations corresponding to multiple modalities. We derive efficient optimization algorithms using the alternating direction method of multipliers (ADMM) to solve the resulting optimization problems. Once the representation coefficients are estimated, the minimum reconstruction rule is used for multimodal recognition. Figure 4.1 gives an overview of the proposed method.

The rest of the Chapter is organized as follows. In Section 4.2, we reviewed related works on multimodal fusion methods. In Section 4.3, we introduce our formulation based on low-rank and joint sparse representation and present two special cases of the proposed method. In Section 4.4, we present an extension of our method

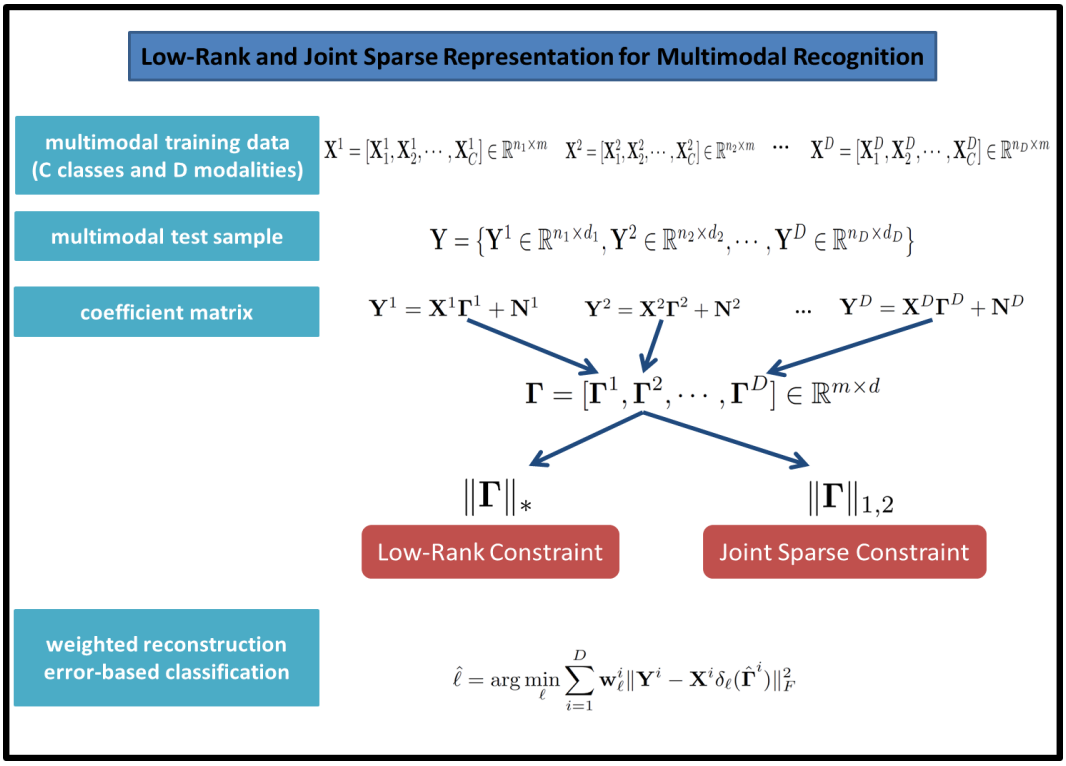


Figure 4.1: An overview of the proposed low-rank and joint sparse representation-based multimodal recognition.

based on common sparse and low-rank representations. An optimization algorithm based on the ADMM method is presented in Section 4.5. Experimental evaluations on three multimodal datasets are presented in Section 4.6. In Section 4.7, the complexity of proposed methods is analyzed. Finally, concluding remarks are presented in Section 4.8 with a brief summary and discussion.

4.2 Related Work

Data fusion can be achieved at several different levels, which can be broadly classified as sensor-level, feature-level, score-level or decision-level fusion. Since feature-level fusion preserves the raw information, it can be more discriminative and robust than score-level or decision-level fusion. The focus of this paper is on designing new feature-level fusion methods and making comparisons with previous feature-level fusion methods.

Differences in features extracted from different modalities in terms of types and dimensions make the feature-level fusion non trivial. One of the simplest methods for feature-level fusion is feature concatenation [65], [66]. However, feature concatenation often tends to be computationally demanding and inefficient. Multiple Kernel Learning (MKL) has also been used to integrate information from multiple features by learning a weighted combination of appropriate kernels. See [67] for more details on various MKL algorithms.

Recent multimodal fusion methods based on sparse or low-rank representations of multimodal data have been shown to produce state-of-the-art results on various

multimodal recognition problems. In [68], a multi-task sparse linear regression model is proposed for image classification. In [69], a joint dynamic sparse representation method was proposed to recognize object viewed from multiple observations (e.g. poses). In [63], a joint sparse representation-based method was proposed for fusing multiple biometrics features. This method is based on multi-task multivariate Lasso [70]. [64] proposed low-rank representation-based multimodal recognition methods. In [71] and [64], the idea of enforcing common sparse (low-rank) representation was shown to be robust and more effective especially when the quality of different modality differs a lot.

In [72], a general collaborative sparse-representation framework for multi-sensor classification is proposed. Joint sparsity is enforced within each sensor's multiple observations and is also simultaneously enforced across heterogeneous sensors. Sparse noise and low rank interference signals are considered in their approach. The objective of the resulting optimization is to seek a joint sparse representation while minimizing the sparse error or low rank interference signals. A multimodal task-driven dictionary learning algorithm with joint sparsity constraint enforced across multiple sources of information is proposed in [73]. In [58], a low-rank and joint sparse representation-based method is proposed for recovering hyperspectral images from a small number of noisy compressive measurements.

Other recent multimodal feature-level fusion methods include [74] and [75]. In [74], a class consistent multi-modal fusion (CCMM) scheme was proposed which essentially extends the application of binary codes [76] for multimodal recognition. In [75], harmonic image fusion was proposed to achieve clutter mitigation and speckle

noise reduction.

4.3 Low-rank and joint sparse representations for multimodal recognition

Suppose we are given a C-class classification problem with D different modalities. Assume that there are m training samples in each modality. For each modality, $i = 1, \cdots, D$, we denote $\mathbf{X}^i = [\mathbf{X}^i_1, \mathbf{X}^i_2, \cdots, \mathbf{X}^i_C]$ as an $n_i \times m$ matrix of training samples containing C sub-matrices $\mathbf{X}^i_j$'s corresponding to C different classes. Each sub-matrix $\mathbf{X}^i_j = [\mathbf{x}^i_{j,1}, \mathbf{x}^i_{j,2}, \cdots, \mathbf{x}^i_{j,m_j}] \in \mathbb{R}^{n_i \times m_j}$ contains a set of training samples from the ith modality corresponding to the jth class. Here, m_j is the number of training samples in class j and n_i is the feature dimension of each sample. As a result, there are in total $m = \sum_{j=1}^{C} m_j$ many samples in $\mathbf{X}^i$. Given a test matrix $\mathbf{Y}$, which consists of D different modalities, $\{\mathbf{Y}^1, \cdots, \mathbf{Y}^D\}$, where each sample $\mathbf{Y}^i$ consists of d_i observations $\mathbf{Y}^i = [\mathbf{y}^i_1, \mathbf{y}^i_2, \cdots, \mathbf{y}^i_{d_i}] \in \mathbb{R}^{n_i \times d_i}$, the objective is to identify the class to which the test sample $\mathbf{Y}$ belongs to.

4.3.1 Basic version

In the case when the data is contaminated by random noise, the observations from ith modality can be modeled as follows

$$\mathbf{Y}^i = \mathbf{X}^i \mathbf{\Gamma}^i + \mathbf{N}^i,$$

where $\mathbf{N}^i$ is small dense additive noise. Let $\mathbf{\Gamma} = [\mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \cdots, \mathbf{\Gamma}^D] \in \mathbb{R}^{m \times d}$ be the coefficient matrix formed by concatenating D representation matrices with $d =$

$\sum_{i=1}^{D} d_i$. We wish to solve for the low-rank and joint sparse matrix $\mathbf{\Gamma}$ by solving the following problem

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i\|_F^2 + \lambda_1\|\mathbf{\Gamma}\|_* + \lambda_2\|\mathbf{\Gamma}\|_{1,2}, \tag{4.1}$$

where $\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}$ is the Frobenius norm of $\mathbf{A}$; $\|\mathbf{A}\|_* = \sum_i \sigma_i(A)$ is the sum of the singular values of $\mathbf{A}$ (i.e. the nuclear norm of $\mathbf{A}$); $\|\mathbf{A}\|_{1,2} = \sum_k \|\mathbf{a}^k\|_2$ and $\mathbf{a}^k$ is the kth row vector of the matrix $\mathbf{A}$ (i.e the row sparsity of $\mathbf{A}$); λ_1 and λ_2 are two positive regularization parameters corresponding to low rank constraint and joint sparse constraint, respectively.

Once the coefficient matrix $\hat{\mathbf{\Gamma}}$ is obtained, the class label associated with an observation vector is declared as the one that produces the smallest approximation error

$$\hat{\ell} = \arg\min_{\ell} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\delta_\ell(\hat{\mathbf{\Gamma}}^i)\|_F^2, \tag{4.2}$$

where $\delta_\ell(\cdot)$ is the matrix indicator function that keeps rows corresponding to the ℓth class and sets all other rows equal to zero.

Ideally, the learned coefficients corresponding to the correct class should exhibit relatively larger values compared to the coefficients corresponding to the incorrect classes. In order to take this assumption into the classification mechanism, for a given coefficient vector obtained from the ith modality, we define $\mathbf{w}_\ell^i$ as:

$$\mathbf{w}^i = \frac{\lambda_1(C\frac{max_\ell\|\delta_\ell(\hat{\mathbf{\Gamma}}^i)\|_*}{\|\hat{\mathbf{\Gamma}}^i\|_*} - 1) + \lambda_2(C\frac{max_\ell\|\delta_\ell(\hat{\mathbf{\Gamma}}^i)\|_{1,2}}{\|\hat{\mathbf{\Gamma}}^i\|_{1,2}} - 1)}{(\lambda_1 + \lambda_2)(C-1)}. \tag{4.3}$$

This weight measures the quality of the learned representation. Representation of high quality will be low-rank ($\max_\ell \|\delta_\ell(\hat{\mathbf{\Gamma}}^i)\|_*$ close to $\|\hat{\mathbf{\Gamma}}^i\|_*$) and will also be joint sparse ($\max_\ell \|\delta_\ell(\hat{\mathbf{\Gamma}}^i)\|_{1,2}$ close to $\|\hat{\mathbf{\Gamma}}^i\|_{1,2}$).

The classification rule (4.2) based on the weighted reconstruction error can be modified as follows

$$\hat{\ell} = \arg\min_{\ell} \sum_{i=1}^{D} \mathbf{w}^i \|\mathbf{Y}^i - \mathbf{X}^i \delta_\ell(\hat{\mathbf{\Gamma}}^i)\|_F^2. \tag{4.4}$$

Similar ideas have been explored in [77] and [63]. We call the resulting algorithm Multimodal Recognition via Low-Rank and Joint Sparse (MRLRJS) representation.

Enforcing joint sparsity (row sparsity) ensures that the number of rows that have nonzero norm to be small. Ideally, these nonzero rows correspond to the true class. A matrix which has row sparsity can also be a low-rank matrix (e.g. many rows of the matrix are null vectors). The reason for enforcing the low-rank constraint is that it can explore the underlying structure of the representation matrix especially in the column sense. For the given input multimodal instance, representations of different modalities are assumed to be correlated, therefore, when these representations are stacked horizontally, the resulting representation matrix is assumed to have a small column rank.

In our experiments, we observed that instances where (4.1) with $\lambda_1 = 0$ fails are often different from those where (4.1) with $\lambda_2 = 0$ fails. Hence, combining the two algorithms may lead to a better multimodal fusion method, since the underlying representation matrix we want to recover is both row-sparse and low-rank simultaneously. Our work is specifically motivated by [59] and [62] where simultaneous ℓ_1-norm and nuclear norm have been studied for general regression and subspace clustering problems, respectively. In contrast, our focus in this paper is specifically on multimodal recognition problems.

4.3.2 Robust version

In the case when data is contaminated by noise and occlusion, the observation from the ith modality can be modeled as follows

$$\mathbf{Y}^i = \mathbf{X}^i\mathbf{\Gamma}^i + \mathbf{N}^i + \mathbf{E}^i,$$

where $\mathbf{N}^i$ is a small dense additive noise and $\mathbf{E}^i$ is a matrix of sparse occlusion with arbitrary large magnitude. By taking advantage of the fact that $\mathbf{E}^i$ is sparse, one can simultaneously estimate $\mathbf{\Gamma}^i$ and $\mathbf{E}^i$ by solving the following optimization problem

$$\begin{aligned}\hat{\mathbf{\Gamma}}, \hat{\mathbf{E}} = \arg\min_{\mathbf{\Gamma},\mathbf{E}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{E}^i\|_F^2 + \lambda_1\|\mathbf{\Gamma}\|_* \\ + \lambda_2\|\mathbf{\Gamma}\|_{1,2} + \lambda_3\sum_{i=1}^{D}\|\mathbf{E}^i\|_1, \end{aligned} \tag{4.5}$$

where $\mathbf{E} = [\mathbf{E}^1, \mathbf{E}^2, \cdots, \mathbf{E}^D]$ is the spare occlusion matrix and $\|\mathbf{A}\|_1 = \sum_{i,j}|A_{i,j}|$ is the ℓ_1-norm of $\mathbf{A}$. Note that $\mathbf{E}$ is just a compact representation and we solve each $\mathbf{E}^i$ separately since their dimensions can be different. Here, λ_1, λ_2 and λ_3 are positive parameters that control the rank of coefficients, joint sparsity of the coefficients and the sparsity of the occlusion term, respectively.

Once $\mathbf{\Gamma}$ and $\mathbf{E}$ are estimated, the effect of occlusion can be removed by setting $\hat{\mathbf{Y}}^i = \mathbf{Y}^i - \hat{\mathbf{E}}^i$. Finally, one can declare the class label associated to an observation vector as

$$\hat{\ell} = \arg\min_{\ell} \sum_{i=1}^{D} \mathbf{w}^i\|\mathbf{Y}^i - \mathbf{X}^i\delta_\ell(\hat{\mathbf{\Gamma}}^i) - \hat{\mathbf{E}}^i\|_F^2, \tag{4.6}$$

where $\mathbf{w}_\ell^i$ is defined in (4.3). We call the resulting algorithm Robust Multimodal Recognition via Low-Rank and Joint Sparse (RMRLRJS) representation.

4.3.3 Two Special Cases

The above formulations take both rank and joint sparsity into consideration and the parameters λ_1 and λ_2 control the relative importance between low-rank and sparse representations, respectively. By selecting λ_1 and λ_2 appropriately, we obtain two special cases of MRLRJS and RMRLRJS.

4.3.3.1 Joint Sparse Representation

If λ_1 is set equal to 0, then the basic and robust versions are simplified as

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i\|_F^2 + \lambda_2 \|\mathbf{\Gamma}\|_{1,2}, \tag{4.7}$$

and

$$\hat{\mathbf{\Gamma}}, \hat{\mathbf{E}} = \arg\min_{\mathbf{\Gamma},\mathbf{E}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{E}^i\|_F^2 + \lambda_2 \|\mathbf{\Gamma}\|_{1,2} + \lambda_3 \sum_{i=1}^{D} \|\mathbf{E}^i\|_1 \tag{4.8}$$

respectively. These simplified formulations essentially correspond to the joint sparse representation-based multimodal fusion algorithms proposed in [63].

4.3.3.2 Low-Rank Representation

If λ_2 is set equal to 0 then, MRLRJS and RMRLRJS are simplified as

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i\|_F^2 + \lambda_1 \|\mathbf{\Gamma}\|_*, \tag{4.9}$$

and

$$\hat{\mathbf{\Gamma}}, \hat{\mathbf{E}} = \arg\min_{\mathbf{\Gamma},\mathbf{E}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{E}^i\|_F^2 + \lambda_1\|\mathbf{\Gamma}\|_* + \lambda_3 \sum_{i=1}^{D} \|\mathbf{E}^i\|_1 \tag{4.10}$$

respectively. These simplified formulations essentially correspond to the multitask multivariate low-rank representations proposed in [64].

4.4 Common low-rank and joint sparse representations for multi-modal recognition

Different from previous formulations, in this section we propose to enforce common sparse and low-rank representations on the coefficients from different modalities. As a result, we are able to exploit the correlations among different modalities better. In this method, the coefficient matrices corresponding to D different modalities are required to be the same as follows

$$\mathbf{\Gamma} = \mathbf{\Gamma}^1 = \mathbf{\Gamma}^2 = \cdots = \mathbf{\Gamma}^D.$$

In order to make the coefficient matrices match in terms of matrix dimensions, for classifying a multi-modal instance in testing phase, the number of samples from each modality has to be the same. With the common representation, low-rank and joint sparse constraint on the concatenated matrix $[\mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \cdots, \mathbf{\Gamma}^D]$ is equivalent to enforcing the constraint on $\mathbf{\Gamma}^1$. Similar ideas have been explored in [78] for image super-resolution and in [64], [71] for multimodal recognition. One of the disadvantages of enfacing such strong constraints is that during the test phase, each modality must have the same number of samples. However, as will be shown later,

such common representation is found to be robust in several biometrics and object recognition applications.

4.4.1 Basic Version

When we consider the common representation, we assume that the ith modality's observations are of the following form

$$\mathbf{Y}^i = \mathbf{X}^i\mathbf{\Gamma} + \mathbf{N}^i.$$

Note that, since the same representation is used for all the modalities in the above model, we let the number of samples from each modality be the same, i.e $\mathbf{Y}^i \in \mathbb{R}^{n_i \times d}$. With this model, the common low-rank and joint sparse representation-based multimodal recognition (MRLRJS-C) problem can be formulated as

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}\|_F^2 + \lambda_1\|\mathbf{\Gamma}\|_* + \lambda_2\|\mathbf{\Gamma}\|_{1,2}. \tag{4.11}$$

Once $\hat{\mathbf{\Gamma}}$ is estimated, it can be used to declare the class label of the observation by the minimum reconstruction error criteria as follows

$$\hat{\ell} = \arg\min_{\ell} \sum_{i=1}^{D} \mathbf{w}\|\mathbf{Y}^i - \mathbf{X}^i\delta_\ell(\hat{\mathbf{\Gamma}})\|_F^2, \tag{4.12}$$

where $\mathbf{w}_\ell$ is defined as

$$\mathbf{w} = \frac{\lambda_1(C\frac{\max_\ell \|\delta_\ell(\hat{\mathbf{\Gamma}})\|_*}{\|\hat{\mathbf{\Gamma}}\|_*} - 1) + \lambda_2(C\frac{\max_\ell \|\delta_\ell(\hat{\mathbf{\Gamma}})\|_{1,2}}{\|\hat{\mathbf{\Gamma}}\|_{1,2}} - 1)}{(\lambda_1 + \lambda_2)(C-1)}. \tag{4.13}$$

4.4.2 Robust Version

In this case, the ith modality's observations are assumed to be of the following form

$$\mathbf{Y}^i = \mathbf{X}^i\mathbf{\Gamma} + \mathbf{N}^i + \mathbf{E}^i.$$

With this, the robust version of the MRLRJS-C (RMRLRJS-C) problem can be formulated as

$$\hat{\mathbf{\Gamma}}, \hat{\mathbf{E}} = \arg\min_{\mathbf{\Gamma},\mathbf{E}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma} - \mathbf{E}^i\|_F^2 + \lambda_1\|\mathbf{\Gamma}\|_* + \lambda_2\|\mathbf{\Gamma}\|_{1,2} + \lambda_3\sum_{i=1}^{D}\|\mathbf{E}^i\|_1. \tag{4.14}$$

Finally, the following minimum reconstruction error rule can be used to classify multimodal data

$$\hat{\ell} = \arg\min_{\ell} \sum_{i=1}^{D} \mathbf{w}\|\mathbf{Y}^i - \mathbf{X}^i\delta_\ell(\hat{\mathbf{\Gamma}}) - \hat{\mathbf{E}}^i\|_F^2, \tag{4.15}$$

where $\mathbf{w}_\ell$ is defined in (4.13).

4.4.3 Two Special Cases

Depending on the selected parameters in (4.14), we obtain the following two special cases.

4.4.3.1 Common Sparse Representation

If λ_1 is set equal to 0, then the basic and the robust versions are simplified as

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2}\sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}\|_F^2 + \lambda_2\|\mathbf{\Gamma}\|_{1,2}, \tag{4.16}$$

and

$$\hat{\mathbf{\Gamma}}, \hat{\mathbf{E}} = \arg\min_{\mathbf{\Gamma},\mathbf{E}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma} - \mathbf{E}^i\|_F^2 + \lambda_2 \|\mathbf{\Gamma}\|_{1,2} + \lambda_3 \sum_{i=1}^{D} \|\mathbf{E}^i\|_1 \tag{4.17}$$

respectively. This formulation is essentially what was proposed in [71].

4.4.3.2 Common Low-Rank Representation

If λ_2 is set equal to 0, then the basic and robust versions are simplified as

$$\hat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}\|_F^2 + \lambda_1 \|\mathbf{\Gamma}\|_*, \tag{4.18}$$

and

$$\hat{\mathbf{\Gamma}}, \hat{\mathbf{E}} = \arg\min_{\mathbf{\Gamma},\mathbf{E}} \frac{1}{2} \sum_{i=1}^{D} \|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma} - \mathbf{E}^i\|_F^2 + \lambda_1 \|\mathbf{\Gamma}\|_* + \lambda_3 \sum_{i=1}^{D} \|\mathbf{E}^i\|_1, \tag{4.19}$$

respectively, which is essentially the common low-rank representation-based multi-modal fusion framework proposed in [64].

4.5 Optimization

In this section, we propose an approach based on the ADMM method [47] to solve the proposed optimization problems. Due to the similarity of these problems, we only provide details on the optimization of (4.5). In ADMM, appropriate auxiliary variables are introduced into the optimization program, the constraints are augmented into the objective function and the Lagrangian is iteratively minimized with respect to the primal variables and maximized with respect to the Lagrange multipliers.

4.5.1 Optimization of RMRLRJS

Problem (4.5) can be reformulated by introducing the auxiliary variables as follows

$$\min_{\mathbf{\Gamma},\mathbf{E},\mathbf{V},\mathbf{U},\mathbf{Z}} \frac{1}{2}\sum_{i=1}^{D}\|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{E}^i\|_F^2 + \lambda_1\|\mathbf{V}\|_* + \lambda_2\|\mathbf{Z}\|_{1,2} + \lambda_3\sum_{i=1}^{D}\|\mathbf{U}^i\|_1 \tag{4.20}$$
$$s.t.\ \mathbf{\Gamma} = \mathbf{V}, \mathbf{\Gamma} = \mathbf{Z}, \mathbf{E}^i = \mathbf{U}^i (i = 1, \cdots, D).$$

Note that similar to $\mathbf{\Gamma}$, we denote $\mathbf{V} = [\mathbf{V}^1, \mathbf{V}^2, \cdots, \mathbf{V}^D] \in \mathbb{R}^{m\times d}$, $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \cdots, \mathbf{Z}^D] \in \mathbb{R}^{m\times d}$ and like $\mathbf{E}$, we let $\mathbf{U} = [\mathbf{U}^1, \mathbf{U}^2, \cdots, \mathbf{U}^D]$ as a compact representation of $\mathbf{U}^i (i = 1, \cdots, D)$ which is, however, solved independently.

Equation (4.20) can be solved using the Augmented Lagrangian Method (ALM) [47]. The augmented Lagrangian function $f_{\alpha_V,\alpha_Z,\alpha_U}(\mathbf{\Gamma}, \mathbf{E}, \mathbf{V}, \mathbf{Z}, \mathbf{U}; \mathbf{A}_V, \mathbf{A}_Z, \mathbf{A}_U)$ is defined as

$$\begin{aligned}
&\min_{\mathbf{\Gamma},\mathbf{E},\mathbf{V},\mathbf{Z},\mathbf{U}} \frac{1}{2}\sum_{i=1}^{D}\|\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i - \mathbf{E}^i\|_F^2 \\
&+ \lambda_1\|\mathbf{V}\|_* + \langle \mathbf{A}_V, \mathbf{\Gamma} - \mathbf{V}\rangle + \frac{\alpha_V}{2}\|\mathbf{\Gamma} - \mathbf{V}\|_F^2 \\
&+ \lambda_2\|\mathbf{Z}\|_{1,2} + \langle \mathbf{A}_Z, \mathbf{\Gamma} - \mathbf{Z}\rangle + \frac{\alpha_Z}{2}\|\mathbf{\Gamma} - \mathbf{Z}\|_F^2 \\
&+ \sum_{i=1}^{D}(\lambda_3\|\mathbf{U}^i\|_1 + \langle \mathbf{A}_U^i, \mathbf{E}^i - \mathbf{U}^i\rangle + \frac{\alpha_U}{2}\|\mathbf{E}^i - \mathbf{U}^i\|_F^2),
\end{aligned} \tag{4.21}$$

where $\mathbf{A}_V$, $\mathbf{A}_Z$ and $\mathbf{A}_U$ are the multipliers of the linear constrains, α_V, α_Z and α_U are the positive parameters, $\langle \mathbf{A}, \mathbf{B}\rangle$ denotes $tr(\mathbf{A}^T\mathbf{B})$. We denote $\mathbf{A}_V = [\mathbf{A}_V^1, \mathbf{A}_V^2, \cdots, \mathbf{A}_V^D] \in \mathbb{R}^{m\times d}$ and $\mathbf{A}_Z = [\mathbf{A}_Z^1, \mathbf{A}_Z^2, \cdots, \mathbf{A}_Z^D] \in \mathbb{R}^{m\times d}$ and $\mathbf{A}_U = [\mathbf{A}_U^1, \mathbf{A}_U^2, \cdots, \mathbf{A}_U^D]$ as a compact representation of $\mathbf{A}_U^i (i = 1, \cdots, D)$.

In the ALM algorithm, $f_{\alpha_V,\alpha_Z,\alpha_U}$ is solved iteratively with respect to $\mathbf{\Gamma}, \mathbf{E}, \mathbf{U}, \mathbf{V}$

and $\mathbf{Z}$ jointly while keeping $\mathbf{A}_V$, $\mathbf{A}_Z$ and $\mathbf{A}_U$ fixed and then updating $\mathbf{A}_V$, $\mathbf{A}_Z$ and $\mathbf{A}_U$ keeping the remaining variables fixed.

Update step for $\mathbf{\Gamma}$

Obtain $\mathbf{\Gamma}_{t+1}$ by minimizing $f_{\alpha_V,\alpha_Z,\alpha_U}$ with respect to $\mathbf{\Gamma}$. This can be done by taking the first-order derivative of $f_{\alpha_V,\alpha_Z,\alpha_U}$ and setting it equal to zero. Furthermore, as the first term of $f_{\alpha_V,\alpha_Z,\alpha_U}$ is a sum of convex functions associated with sub-matrices $\mathbf{\Gamma}^i$, one can find $\mathbf{\Gamma}^i_{t+1}(i = 1, \cdots, D)$ by solving the following linear system

$$(\mathbf{X}^{iT}\mathbf{X}^i + (\alpha_V + \alpha_Z)\mathbf{I})\mathbf{\Gamma}^i_{t+1} = \mathbf{X}^{iT}(\mathbf{Y}^i - \mathbf{E}^i_t) + \alpha_V\mathbf{V}^i_t - \mathbf{A}^i_{V,t} + \alpha_Z\mathbf{Z}^i_t - \mathbf{A}^i_{Z,t}, \quad (4.22)$$

where $\mathbf{I}$ is $m \times m$ identity matrix and $\mathbf{E}^i_t$, $\mathbf{V}^i_t$, $\mathbf{Z}^i_t$, $\mathbf{A}^i_{V,t}$ and $\mathbf{A}^i_{Z,t}$ are submatrices of $\mathbf{E}_t, \mathbf{V}_t, \mathbf{Z}_t$, $\mathbf{A}_{V,t}$ and $\mathbf{A}_{Z,t}$, respectively. When m is is not very large, one can simply apply matrix inversion to obtain $\mathbf{\Gamma}^i_{t+1}$ from Eq.(4.22). For large values of m, gradient-based methods should be employed to obtain $\mathbf{\Gamma}^i_{t+1}$.

Update step for $\mathbf{E}$

The second subproblem is similar to the first in nature and $\mathbf{E}^i_{t+1}(i = 1, \cdots, D)$ can be obtained as

$$\mathbf{E}^i_{t+1} = (1 + \alpha_U)^{-1}(\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}^i_{t+1} + \alpha_U\mathbf{U}^i_t - \mathbf{A}^i_{U,t}),$$

where $\mathbf{U}^i_t$ and $\mathbf{A}^i_{U,t}$ are submatrices of $\mathbf{U}_t$ and $\mathbf{A}_{U,t}$, respectively.

Update step for $\mathbf{U}$

In order to update $\mathbf{U}^i_{t+1}(i = 1, \cdots, D)$, one needs to solve the following ℓ_1 minimization problem

$$\min \frac{1}{2}\|\mathbf{E}^i_{t+1} + \alpha_U^{-1}\mathbf{A}^i_{U,t} - \mathbf{U}^i\|_F^2 + \frac{\lambda_3}{\alpha_U}\|\mathbf{U}^i\|_1, \tag{4.23}$$

whose solution is given by [79]

$$\mathbf{U}^i_{t+1} = \mathcal{S}\left(\mathbf{E}^i_{t+1} + \alpha_U^{-1}\mathbf{A}^i_{U,t}, \frac{\lambda_3}{\alpha_U}\right),$$

where $\mathcal{S}(a, b) = sgn(a)(|a| - b)$ for $|a| \geq b$ and zero otherwise.

Update step for $\mathbf{V}$

The subproblem for updating $\mathbf{V}$ has the following form

$$\min \frac{1}{2}\|\mathbf{\Gamma}_{t+1} + \alpha_V^{-1}\mathbf{A}_{V,t} - \mathbf{V}\|_F^2 + \frac{\lambda_1}{\alpha_V}\|\mathbf{V}\|_*. \tag{4.24}$$

Solution to this optimization problem is obtained by shrinking the singular values of $\mathbf{\Gamma}_{t+1} + \alpha_V^{-1}\mathbf{A}_{V,t}$ [80], [81]. As a result, we obtain the following update for $\mathbf{V}$

$$\mathbf{V}_{t+1} = \mathbf{F}\mathcal{S}(\mathbf{\Sigma}, \frac{\lambda_1}{\alpha_V})\mathbf{B}^T,$$

where $\mathbf{F}\mathbf{\Sigma}\mathbf{B}^T$ is the Singular Value Decomposition (SVD) of $\mathbf{\Gamma}_{t+1} + \alpha_V^{-1}\mathbf{A}_{V,t}$. Same $\mathcal{S}(a, b)$ is applied as above.

Update step for $\mathbf{Z}$

In order to update $\mathbf{Z}$, we need to solve the following optimization problem

$$\min \frac{1}{2}\|\mathbf{\Gamma}_{t+1} + \alpha_Z^{-1}\mathbf{A}_{Z,t} - \mathbf{Z}\|_F^2 + \frac{\lambda_2}{\alpha_Z}\|\mathbf{Z}\|_{1,2}. \tag{4.25}$$

Due to the separable structure of (4.25), it can be solved by minimizing it with respect to each row of $\mathbf{Z}$ separately. Following the method used in [63], we let $\gamma_{i,t+1}$, $\mathbf{a}_{Z_{i,t}}$ and $\mathbf{z}_{i,t+1}$ be the ith row of matrices $\mathbf{\Gamma}_{t+1}$, $\mathbf{A}_{Z,t}$ and $\mathbf{Z}_{t+1}$ respectively. Then for each row, we solve the following subproblem

$$\mathbf{z}_{i,t+1} = \min \frac{1}{2}\|\mathbf{p} - \mathbf{z}\|_2^2 + \frac{\lambda_2}{\alpha_Z}\|\mathbf{z}\|_2, \tag{4.26}$$

where $\mathbf{p} = \gamma_{i,t+1} + \mathbf{a}_{Z_{i,t}}\alpha_Z^{-1}$. The closed form solution of (4.26) is given as follows

$$\mathbf{z}_{i,t+1} = \left(\mathbf{1} - \frac{\lambda_\mathbf{2}}{\alpha_\mathbf{Z}\|\mathbf{p}\|_\mathbf{2}}\right)_+ \mathbf{p},$$

where $(\mathbf{v})_+$ is the vector with entries receiving values $\max(v_i, 0)$.

Update steps for $\mathbf{A}_V$, $\mathbf{A}_Z$ and $\mathbf{A}_U^i (i = 1, \cdots, D)$

Finally, the Lagrange multipliers are updated as

$$\mathbf{A}_{V,t+1} = \mathbf{A}_{V,t} + \alpha_V(\mathbf{\Gamma}_{t+1} - \mathbf{V}_{t+1}), \tag{4.27}$$

$$\mathbf{A}_{Z,t+1} = \mathbf{A}_{Z,t} + \alpha_Z(\mathbf{\Gamma}_{t+1} - \mathbf{Z}_{t+1}), \tag{4.28}$$

$$\mathbf{A}_{U,t+1}^i = \mathbf{A}_{U,t}^i + \alpha_U(\mathbf{E}_{t+1}^i - \mathbf{U}_{t+1}^i). \tag{4.29}$$

The proposed ADMM algorithm for solving the RMRLRJS problem is summarized in Algorithm 3. Note that the optimization problem is not convex and there does not exist any guarantee for the Algorithm 3 to converge. The convergence issue of ADMM is still not fully understood and remains an open research problem. Yet, ADMM works well in practice. For our proposed methods, the termination condition is either when the difference of the cost function errors is below some threshold or the maximum number of iteration is reached.

Algorithm 3: Robust Multimodal Recognition via Low-Rank and Joint Sparse Representation (RMRLRJS)

Input: Multi-modal training samples $\{\mathbf{X}_i\}_{i=1}^D$, test sample $\{\mathbf{Y}_i\}_{i=1}^D$, $\lambda_1, \lambda_2, \lambda_3$, α_V, α_Z and α_U

Initialization:

$\mathbf{\Gamma}_0, \mathbf{V}_0, \mathbf{Z}_0, \mathbf{U}_0, \mathbf{A}_{V,0}, \mathbf{A}_{Z,0}, \mathbf{A}_{U,0}$ are initialized to be zero matrices.

Until convergence do

1. Update $\mathbf{\Gamma}$: $\mathbf{\Gamma}_{t+1} = [\mathbf{\Gamma}_{t+1}^1, \cdots, \mathbf{\Gamma}_{t+1}^D]$, where

$$\mathbf{\Gamma}_{t+1}^i = ({\mathbf{X}^i}^T\mathbf{X}^i + (\alpha_V + \alpha_Z)\mathbf{I})^{-1}({\mathbf{X}^i}^T(\mathbf{Y}^i - \mathbf{E}_t^i) + \alpha_V\mathbf{V}_t^i - \mathbf{A}_{V,t}^i + \alpha_Z\mathbf{Z}_t^i - \mathbf{A}_{Z,t}^i)$$

2. Update $\mathbf{E}$: $\mathbf{E}_{t+1} = [\mathbf{E}_{t+1}^i, \cdots, \mathbf{E}_{t+1}^D]$, where

$$\mathbf{E}_{t+1}^i = (1 + \alpha_E)^{-1}(\mathbf{Y}^i - \mathbf{X}^i\mathbf{\Gamma}_{\boldsymbol{t+1}}^{\boldsymbol{i}} + \alpha_U\mathbf{U}_t^i - \mathbf{A}_{U,t}^i)$$

3. Update $\mathbf{U}$: $\mathbf{U}_{t+1} = [\mathbf{U}_{t+1}^i, \cdots, \mathbf{U}_{t+1}^D]$, where

$$\mathbf{U}_{t+1}^i = \mathcal{S}\left(\mathbf{E}_{t+1}^i + \alpha_U^{-1}\mathbf{A}_{U,t}^i, \frac{\lambda_3}{\alpha_U}\right)$$

4. Update $\mathbf{V}$:

$$\mathbf{V}_{t+1} = \mathbf{F}\mathcal{L}_{\frac{\lambda_1}{\alpha_V}}(\mathbf{\Sigma})\,\mathbf{B}^T$$

5. Update $\mathbf{Z}$:

$$\mathbf{z}_{i,t+1} = \left(\mathbf{1} - \frac{\lambda_{\mathbf{2}}}{\alpha_{\mathbf{Z}}\|\mathbf{p}\|_{\mathbf{2}}}\right)_+\mathbf{p}$$

6. Update $\mathbf{A}_V$, $\mathbf{A}_Z$, $\mathbf{A}_U^i (i = 1, \cdots, D)$:

$$\mathbf{A}_{V,t+1} = \mathbf{A}_{V,t} + \alpha_V(\mathbf{\Gamma}_{t+1} - \mathbf{V}_{t+1})$$

$$\mathbf{A}_{Z,t+1} = \mathbf{A}_{Z,t} + \alpha_Z(\mathbf{\Gamma}_{t+1} - \mathbf{Z}_{t+1})$$

$$\mathbf{A}_{U,t+1}^i = \mathbf{A}_{U,t}^i + \alpha_U(\mathbf{E}_{t+1}^i - \mathbf{U}_{t+1}^i)$$

Classification:

Let $\hat{\mathbf{E}}^i = \mathbf{E}_{t+1}^i (i = 1, \cdots, D)$ and $\hat{\mathbf{\Gamma}} = \mathbf{\Gamma}_{t+1}$,

1. Compute weight $\mathbf{w}_\ell^i$ by (4.3)

2. Assign the class label with minimum error:

$$\hat{\ell} = \arg\min_\ell \sum_{i=1}^D \mathbf{w}_\ell^i \|\mathbf{Y}^i - \mathbf{X}^i\delta_\ell(\hat{\mathbf{\Gamma}}^i) - \hat{\mathbf{E}}^i\|_F^2$$

Output: class label $\hat{\ell}$

Figure 4.2: Sample fingerprint and iris images from the WVU dataset.

4.5.2 Optimization of RMRLRJS-C

The RMRLRJS-C problem (4.14) can be optimized in a similar way using the ADMM method. However, there are a few key differences in solving the subproblems. In particular, $\mathbf{\Gamma}$ is not separated into D different parts and $\mathbf{\Gamma}$ can be updated as

$$\mathbf{\Gamma}_{t+1} = (\sum_{i=1}^{D} \mathbf{X}^{i^T}\mathbf{X}^i + (\alpha_V + \alpha_Z)\mathbf{I})^{-1}(\sum_{i=1}^{D} \mathbf{X}^{i^T}(\mathbf{Y}^i - \mathbf{E}^i) + \alpha_V \mathbf{V}_t^i - \mathbf{A}_{V,t}^i + \alpha_Z \mathbf{Z}_t^i - \mathbf{A}_{Z,t}^i).$$

After solving $\hat{\mathbf{E}}_i (i = 1, \cdots, D)$ and $\hat{\mathbf{\Gamma}}$, the class label can by obtained by using (4.13) and (4.15).

4.6 Experimental Results

In this section, we evaluate the proposed algorithms on three publicly available multimodal recognition datasets, namely the WVU multimodal biometrics dataset [12], UMDAA multimodal active authentication dataset [10], [11] and multimodal object recognition [13]. We compare the proposed method with various feature-level fusion methods including multiple kernel learning based multi-modal fusion

method (MKL) [82], joint sparse representation-based multi-modal fusion methods (SMBR-WE and SMBR-E) [63], common sparse representation-based multi-modal fusion methods (MCSR and RMCSR) [71], low-rank representation-based multi-modal fusion methods (MLRR, RMLRR, MCLRR and RMCLRR) [64] and the class consistent multi-modal fusion (CCMM) [74].

The proposed methods can have up to six parameters during the optimization procedure. To efficiently tune these parameters, we adopt the following strategy: solve for appropriate parameters for joint sparse representation-based optimization and low-rank representation-based optimization separately and then weight these parameters to control their relative contributions to the final recognition. For example, in order to tune the parameters in Algorithm 3, we first consider the sparsity constraint only by letting λ_1 be 0 and obtain "optimal" λ_2 and λ_{3_s}, α_Z and α_{U_s} through grid search. Then, we consider the low-rank constraint only and obtain λ_1 and λ_{3_r}, α_V and α_{U_r}. Finally, we introduce a parameter $r(0 \leq r \leq 1)$ to control the relative contribution and the final parameters used are $r\lambda_1$, $(1-r)\lambda_2$ and $r\lambda_{3_r} + (1-r)\lambda_{3_s}$, $r\alpha_V$, $(1-r)\alpha_Z$ and $r\alpha_{U_r} + (1-r)\alpha_{U_s}$.

4.6.1 WVU multimodal biometrics dataset

The WVU biometrics dataset is a comprehensive collection of different biometric modalities such as fingerprint, iris, palmprint, hand geometry, and voice from subjects of different age, gender, and ethnicity. It is a challenging dataset as many of these samples are corrupted with blur, occlusion, and sensor noise. Following the

experimental settings described in [63], we choose four fingerprint modalities and two iris modalities on a subset of 219 subjects having data in all these modalities. Figure 4.2 shows some sample fingerprint and iris images from this dataset.

Preprocessing and Feature Extraction

We applied the same preprocessing and feature extraction methods used in [63]. In particular, fingerprint images were enhanced using the filtering methods described in [83]. After detecting the core point [84], Gabor features were extracted around the core point and a feature vector of dimension 3600 was obtained for each fingerprint image. The iris images were segmented using the method proposed in [85] and the publicly available code described in [86] was applied to create 25×240 iris templates. A Gabor feature of dimension 6000 was generated for every iris image.

Experiment Setup, Results and Analysis

The data instances (one instance includes six samples corresponding to six modalities) were randomly divided into four training instances per class and the remaining instances were used for testing. As a result, 876 instances were used for training and 519 instances were used for testing. The recognition result was averaged over five runs and we report the mean and standard deviation of rank one recognition accuracy. The rank one recognition results comparing the proposed methods with other feature-level multimodal fusion methods are shown in Table 4.1 and Table 4.2 for each modality alone and fusion of modalities, respectively. RMRLRJS-C shows

the best recognition performance and the corresponding parameters λ_1, λ_2, λ_3, $r\alpha_V$, α_U, α_Z are set equal to 0.0004, 0.0006, 0.0007, 0.0004, 0.0064, 0.006, respectively.

Methods	Finger 1	Finger 2	Finger 3	Finger 4	Iris 1	Iris 2
CCMM	67.8 ± 1.2	86.9 ± 1.1	69.4 ± 1.9	89.3 ± 1.6	60.5 ± 1.7	61.2 ± 0.9
SMBR-WE	68.1 ± 1.1	88.4 ± 1.2	69.2 ± 1.5	87.5 ± 1.5	60.0 ± 1.5	62.1 ± 0.4
SMBR-E	67.1 ± 1.0	87.9 ± 0.8	67.4 ± 1.9	86.9 ± 1.5	62.5 ± 1.2	64.3 ± 1.0
MCSR	70.3 ± 1.0	90.1 ± 0.8	69.2 ± 2.3	89.5 ± 1.4	62.6 ± 1.8	64.6 ± 1.0
RMCSR	69.8 ± 1.4	89.4 ± 1.0	69.2 ± 2.3	89.2 ± 1.1	**70.5 ± 1.1**	**71.7 ± 0.5**
MLRR	70.0 ± 1.8	90.0 ± 0.9	68.3 ± 1.8	89.6 ± 1.4	59.0 ± 1.8	60.1 ± 0.8
RMLRR	**70.4 ± 1.5**	89.8 ± 1.0	68.8 ± 2.1	89.9 ± 1.9	63.0 ± 1.4	65.2 ± 0.6
MCLRR	68.5 ± 1.9	88.8 ± 1.2	67.5 ± 1.5	88.5 ± 1.6	56.5 ± 1.4	58.8 ± 0.6
RMCLRR	68.5 ± 1.5	88.3 ± 1.1	67.0 ± 1.6	87.9 ± 1.7	58.7 ± 1.0	60.1 ± 0.6
MRLRJS	69.7 ± 1.1	89.7 ± 1.3	70.6 ± 1.6	90.4 ± 0.6	59.6 ± 1.0	61.0 ± 0.4
RMRLRJS	68.6 ± 1.3	89.3 ± 1.1	69.0 ± 2.0	89.0 ± 1.4	63.5 ± 1.1	64.6 ± 1.0
MRLRJS-C	69.5 ± 0.9	90.0 ± 1.0	70.1 ± 1.6	90.4 ± 0.5	59.1 ± 0.8	60.6 ± 0.5
RMRLRJS-C	70.1 ± 1.8	**90.1 ± 0.3**	**71.2 ± 1.3**	**90.5 ± 0.1**	69.5 ± 1.3	69.8 ± 0.6

Table 4.1: Rank one recognition accuracy (in %) for WVU biometric multi-modal dataset for individual modality.

From the results shown in Table 4.1 and Table 4.2, we make the following observations: (1) All the considered methods achieve better recognition accuracy when fusing multiple modalities than using a single modality for recognition. (2) Robust formulations by including the sparse error term in the optimization can lead to better recognition results. (3) Compared to applying the low-rank constraint or joint sparsity constraint alone, the proposed methods that enforce both low-

Methods	4 Fingerprints	2 Irises	All Modalities
MKL	86.2 ± 1.2	76.8 ± 2.5	89.8 ±0.9
CCMM	**98.9**± 0.5	82.9 ± 1.4	99.6 ±0.2
SMBR-WE	97.9 ± 0.4	76.5 ± 1.6	98.7 ±0.2
SMBR-E	97.6 ± 0.6	78.2 ± 1.2	98.6 ±0.5
MCSR	95.6 ± 0.4	78.3 ± 0.2	98.2 ±0.4
RMCSR	96.1 ± 0.6	85.3 ± 1.9	99.4 ±0.5
MLRR	98.7 ± 0.6	74.0 ± 0.9	98.9 ±0.4
RMLRR	98.7 ± 0.5	78.2 ± 1.2	99.1 ±0.4
MCLRR	96.0 ± 0.4	74.9 ± 1.7	98.6 ±0.5
RMCLRR	96.5 ± 0.2	77.0 ± 1.6	99.4 ±0.5
MRLRJS	98.5 ± 0.7	75.9 ± 0.9	99.0 ± 0.2
RMRLRJS	98.2 ± 0.5	78.6 ± 1.7	99.2 ± 0.1
MRLRJS-C	96.0 ± 0.6	76.2 ± 2.12	99.0 ± 0.7
RMRLRJS-C	96.6 ± 0.2	**85.6 ± 1.7**	**99.8 ± 0.1**

Table 4.2: Rank one recognition accuracy (in %) for the WVU multimodal biometric dataset for fusion of different modalities.

rank and joint sparse constraints perform better. (4) Common representation-based methods RMRLRJS-C perform slightly better than their corresponding methods without applying common representation constraints.

For the first proposed formulation (MRLRJS and RMRLRJS), the representation we seek is $\mathbf{\Gamma} = [\mathbf{\Gamma}^1, \mathbf{\Gamma}^2, \cdots, \mathbf{\Gamma}^D] \in \mathbb{R}^{m \times d}$. The advantage of this formulation is that the information from each modality is preserved in the representation matrix; the disadvantage is that a single modality may determine the low-rank and joint sparse property of the representation matrix, thus determine the overall performance. For example, if the representation of a certain modality is not low-rank or joint sparse, this modality can still determine the overall low-rank and joint sparse property of the overall representation matrix and as a result, we may get a poor performance.

For the second proposed formulation (MRLRJS-C and RMRLRJS-C), the representation is the same for all D modalities, i.e. $\mathbf{\Gamma} = \mathbf{\Gamma}^1 = \mathbf{\Gamma}^2 = \cdots = \mathbf{\Gamma}^D$. The advantage of this formulation is that it satisfies the low-rank and joint sparse constraint more easily and it is more robust as each modality contributes partially to the same representation and no modality can determine the overall representation alone; the disadvantage is that it loses some discriminative information since only a single representation is enforced for all modalities.

Therefore for this dataset in which the performance of each modalities is on the same level, the proposed methods work. However, due to the advantage and disadvantage of common representation (second formulation), RMRLRJS-C works only slightly better than RMRLRJS as we see in observation (4).

4.6.2 UMDAA Dataset

Experimental Setup, Results and Analysis

In order to evaluate the proposed multimodal fusion methods, we sampled a subset from this dataset. For each user in each of the three sessions, thirty face images and thirty touch swipes were randomly selected and the resulting subset consisted of 4500 face images and 4500 touch swipes corresponding to 50 users across three sessions. We selected 10, 15 and 20 instances for each user to form the training data, and use the remaining data for testing. In total, there are 500, 750, and 1000 instances for training and 4000, 3750 and 3500 instances for testing. Each instance contains a 504-dimensional feature vector for the face image and a 27-dimensional feature vector for screen touch gestures. By randomly splitting the data for training and testing, we repeated each experiment ten times and report the mean and standard deviation of the rank one recognition accuracy.

The reason why we choose a small fraction of data for training is because in active authentication, the matching algorithm is supposed to work on mobile devices nearly in real-time. Our algorithm calculates the representation (either sparse or low rank or both) using the training samples, thus more training samples means high-dimensional representation and high computational cost, which should be tuned carefully in order to achieve a balance between performance and speed.

The experimental results comparing our proposed methods with other fusion methods are shown in Table 4.3 and Table 4.4, and Table 4.5 respectively, when

Methods	Face	Touch	Face & Touch
MKL	72.58 ± 1.08	**36.02 ± 0.49**	75.13 ±2.22
CCMM	76.87 ± 1.18	33.54 ± 1.71	79.25 ±1.39
SMBR-WE	75.37 ± 1.13	30.40 ± 1.59	66.69 ±0.78
SMBR-E	73.05 ± 1.29	27.72 ± 1.50	64.49 ±1.61
MCSR	78.23 ± 0.98	28.44 ± 1.27	78.50 ±0.87
RMCSR	78.38 ± 0.87	27.72 ± 1.50	78.44 ±0.87
MLRR	76.04 ± 0.92	21.95 ± 1.41	69.24 ±0.85
RMLRR	75.94 ± 1.16	21.88 ± 1.35	69.21 ±1.17
MCLRR	75.49 ± 1.03	22.02 ± 1.37	78.58 ±1.21
RMCLRR	72.72 ± 1.49	21.88 ± 1.34	77.93 ±1.35
MRLRJS	77.36 ± 1.19	31.09 ± 1.61	68.96 ±0.86
RMRLRJS	77.15 ± 0.98	28.82 ± 1.64	63.74 ±1.04
MRLRJS-C	**80.28 ± 1.01**	23.85 ± 1.57	**81.94 ± 1.09**
RMRLRJS-C	78.77 ± 1.05	24.95 ± 1.56	81.15 ± 1.05

Table 4.3: Rank one recognition accuracy (in %) for different fusion methods using 10 samples from each user for training.

we use 10, 15 and 20 training instances for each user. MRLRJS-C yielded the best recognition performance and the corresponding parameters λ_1, λ_2, $r\alpha_V$, α_Z are set equal to 0.0014, 0.0001, 0.45, 0.001, respectively.

From the results shown in Tables 4.3, 4.4 and 4.5, we make the following observations: (1) Face modality works much better than touch modality. (2) As we increase the number of training samples, we observe consistent performance for each fusion method. The more training samples, the better each method perform

Methods	Face	Touch	Face & Touch
MKL	77.23 ± 0.57	**39.19 ± 1.25**	80.80 ±1.22
CCMM	79.78 ± 0.61	37.27 ± 1.11	83.16 ±1.03
SMBR-WE	81.44 ± 0.49	32.42 ± 1.13	74.31 ±1.10
SMBR-E	79.12 ± 0.61	30.18 ± 1.22	71.90 ±1.36
MCSR	83.71 ± 0.47	29.79 ± 1.14	84.95 ±0.49
RMCSR	83.96 ± 0.45	29.93 ± 1.14	85.02 ±0.43
MLRR	81.04 ± 0.60	23.26 ± 1.57	75.82 ±1.06
RMLRR	81.19 ± 0.63	23.27 ± 1.69	76.28 ±1.06
MCLRR	80.60 ± 0.52	23.26 ± 1.58	83.68 ±0.53
RMCLRR	79.19 ± 0.72	23.27 ± 1.65	83.75 ±0.66
MRLRJS	83.09 ± 0.61	32.64 ± 1.18	76.08 ±1.02
RMRLRJS	81.54 ± 0.63	31.21 ± 1.34	71.28 ±0.99
MRLRJS-C	**85.47 ± 0.54**	24.78 ± 1.43	**87.45 ± 0.58**
RMRLRJS-C	84.44 ± 0.38	25.94 ± 1.28	87.26 ± 0.46

Table 4.4: Rank one recognition accuracy (in %) for different fusion methods using 15 samples from each user for training.

Methods	Face	Touch	Face & Touch
MKL	78.36 ± 0.94	**41.48 ± 0.56**	82.20 ±0.61
CCMM	83.29 ± 0.71	40.15 ± 1.03	87.54 ±0.72
SMBR-WE	85.83 ± 0.66	32.71 ± 0.99	74.64 ±0.85
SMBR-E	87.47 ± 0.66	28.61 ± 1.45	74.88 ±1.00
MCSR	87.06 ± 0.64	29.07 ± 1.07	88.49 ±0.95
RMCSR	87.11 ± 0.71	29.08 ± 1.16	88.48 ±0.56
MLRR	87.67 ± 0.70	23.35 ± 0.99	78.94 ±0.78
RMLRR	88.02 ± 0.82	23.52 ± 1.07	79.65 ±0.86
MCLRR	87.44 ± 0.73	23.41 ± 1.10	89.33 ±0.61
RMCLRR	86.69 ± 0.85	23.61 ± 1.11	89.60 ±0.85
MRLRJS	86.30 ± 0.74	33.97 ± 1.13	80.66 ± 0.86
RMRLRJS	85.64 ± 0.78	32.01 ± 1.19	75.80 ± 0.88
MRLRJS-C	**88.58 ± 0.60**	26.78 ± 1.17	90.42 ± 0.54
RMRLRJS-C	87.57 ± 0.68	26.64 ± 1.11	**90.45 ± 0.62**

Table 4.5: Rank one recognition accuracy (in %) for different fusion methods using 20 samples from each user for training.

in terms of both single modality and the fusion of two modalities. (3) Methods without enforcing common representation failed to fuse face and touch modality to generate better performance than using single modality alone. On contrary, methods enforcing common representation (MCSR, RMCSR, MCLRR, RMCLRR, MRLRJS-C, RMRLRJS-C) successfully fused the two modalities.

In this dataset, faces (strong modality) as physical biometrics are more robust and reliable while screen touch gestures (weak modality), as a kind of behavioral biometric, exhibit more variations and can changes more easily. The performance of face modality and touch modality differs a lot. For fusion methods enforcing a common representation, it is more robust as each modality contributes partially to the same representation and no modality can determine the overall representation alone. Therefore, it can successfully fuse two modalities even with presence of weak modality. However, for fusion methods without enforcing common representation, weak modality can significantly influence the quality of the overall representation and lead to worse performance when fusing two modalities compared to using face modality alone.

4.6.3 Pascal-Sentence Dataset

Pascal-Sentence dataset is a multimodal dataset consisting of two modalities, i.e, image and sentences describing the image [13]. The images are chosen from the PASCAL VOC 2008 Challenge, which is a benchmark dataset for object recognition and detection. 1000 images were randomly selected from 20 classes. Each image

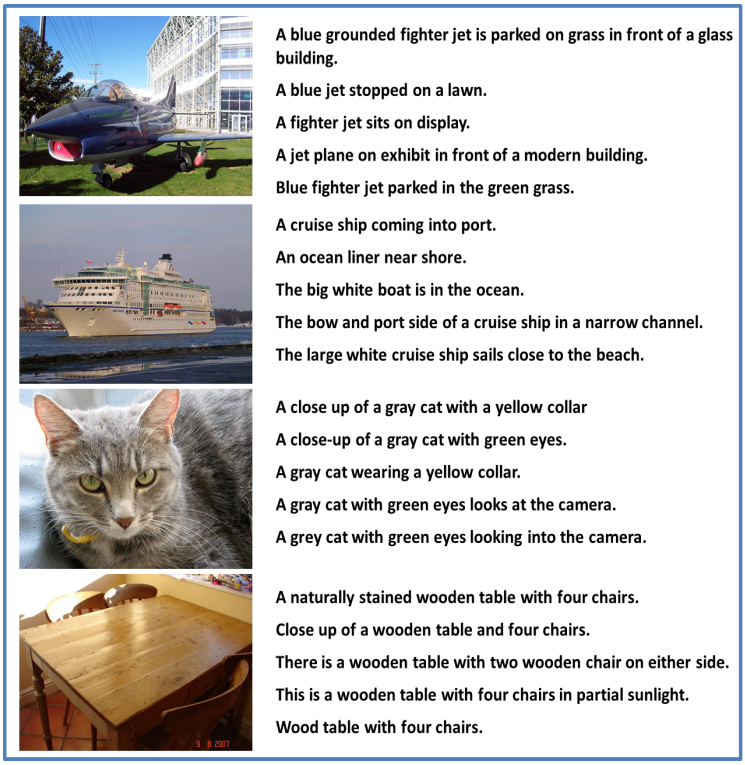


Figure 4.3: Sample images and corresponding sentences from the Pascal-Sentence dataset.

was annotated with five sentences using Amazon's Mechanical Turk. Samples images and the corresponding sentences from this dataset are shown in Figure 4.3.

Preprocessing and Feature Extraction

We follow the same feature extraction method as described in [74]. Specifically, the image features are collections of responses from a variety of detectors, image classifiers and scene classifiers. The semantic features were constructed by using word-net semantic with a dictionary of 1200 words. The details of feature extraction for both modalities are described in [87]. These low-level features were then

Methods	Intensity Features	Semantic Features	Fusion
MKL	67.2	64.4	76
CCMM	66.2	63.2	77.2
SMBR	66.2	69.6	75.4
MRLRJS	**75.5 ± 0.2**	**77.7 ± 0.1**	**82.7 ± 0.3**
RMRLRJS	**75.5 ± 0.2**	**77.7 ± 0.1**	**82.7 ± 0.3**
MRLRJS-C	75.0 ± 0.2	74.6 ± 0.5	81.1 ±0.6
RMRLRJS-C	75.0 ± 0.2	74.6 ± 0.5	81.1 ±0.6

Table 4.6: Classification accuracy (in %) for the Pascal-Sentence dataset.

converted to binary codes using the methods described in [76]. The binary codes were then used to evaluate the performance of various feature-level fusion methods.

Experimental Setup, Results and Analysis

Following the experimental setup in [74], we randomly chose 500 samples for training and kept the remaining 500 samples for testing and calculated the performance of our method. We repeated this process five times and report the final accuracy in terms of mean and standard deviation (std) in Table 4.6. Note that the results of the other methods are directly copied from [74] which essentially follows the same protocol but does not report the std values. RMRLRJS yielded the best recognition performance and the corresponding parameters λ_1, λ_2, λ_3, $r\alpha_V$, α_Z, α_U, are set equal to 0.5, 1, 0.5, 0.5, 1, 0.5, respectively.

From the results shown in Tables 4.6, we make the following observations: (1) The performance of each modality is on the same level. (2) The robust version of

each formulations (RMRLRJS, EMRLRJS-C) did not yield improved performance than their corresponding basic version (MRLRJS, MRLRJS-C). (3) Enforcing a common representation did not yield improved performance.

In this dataset, since the performance of each modality is similar, both formulations perform comparably. The proposed formulation enforcing common representation does not show better results because we get a more robust representation at the cost of losing (discriminative) information. Also, the robust version of each formulation does not show significant performance because of the fact that the original low-level features were converted into binary codes which are already robust to sparse errors.

4.6.4 Low-Rank versus Joint Sparsity

To study the relative contribution of low-rank constraint and joint sparse constraint, we vary the parameter r from 0 to 1 in the increments of 0.1 and plot the mean rank one recognition accuracy for RMRLRJS-C. When $r = 0$, our method reduces to RMCSR and when $r = 1$ the proposed method reduces to RMCLRR. Figure 4.4 shows the performance change of RMRLRJS-C under different values of r. This figure clearly illustrates the advantage of enforcing low-rank and joint sparsity constraints together over enforcing just low-rank or joint sparsity constraint alone.

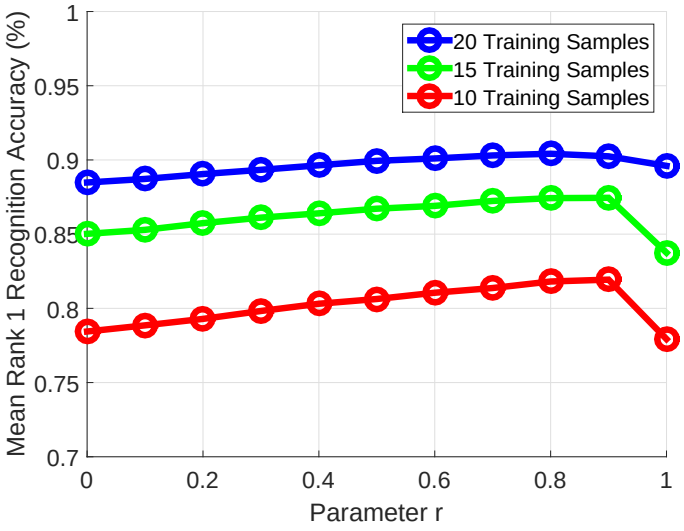


Figure 4.4: Mean rank one recognition accuracy versus the relative contribution of low-rank and joint sparsity constraint.

4.6.5 Weighted vs Non-Weighted Classification

We applied the weighted reconstruction error to assign a given test instance after solving the (common) low-rank and joint sparse representation. To empirically compare these two classification strategies, we applied non-weighted classification using the same representation obtained by the proposed methods on the three datasets and report the recognition. As shown in Table 4.7, the weighted classification rule provides no worse results than those obtained by non-weighted classification.

4.7 Complexity Analysis

To analyze the computational complexity of the proposed methods, we look at each step in the algorithm. For simplicity, we assume the number of modalities

Dataset	Non-Weighted	Weighted
WVU	99.80	99.80
UMDAA	89.51	90.45
Pascal-Sentence	81.48	82.72

Table 4.7: Rank one recognition accuracy (in %) for weighted and non-weighted classification on three datasets.

is D, the number of classes is C, the dimension of the feature vector from different modality is n, the number of training samples is m, the number of iterations is k and the number of observations from different modality in one test sample is p. D and p are usually much smaller than C, m and n. k depends on how quickly the algorithm can converge.

In general, the complexity of matrix multiplication is $\mathcal{O}(n^3)$ and the complexity of matrix addition is $\mathcal{O}(n^2)$ for two $n \times n$ matrices. The complexity of matrix inversion and singular value decomposition is $\mathcal{O}(n^3)$ for an $n \times n$ matrix. For the proposed algorithm, in every iteration, the complexity of computing $\mathbf{\Gamma}$ and $\mathbf{E}$ is $\mathcal{O}(mnpD)$. Note that the matrix inversion part can be computed in advance since it is fixed. Computing $\mathbf{U}$ requires thresholding each element and its complexity is $\mathcal{O}(npD)$. Computing $\mathbf{V}$ involves singular value decomposition, singular value shrinking and matrix multiplication and their complexity is $\mathcal{O}(m^2pD)$. The complexity of computing $\mathbf{Z}$ is $\mathcal{O}(mpD)$. The complexity of computing $\mathbf{A}_V$, $\mathbf{A}_Z$, $\mathbf{A}_U$ is also $\mathcal{O}(mpD)$. Therefore, computing the coefficient matrix through k iterations

requires the computations in the order of $\mathcal{O}(k(mnpD + m^2pD))$. For classifying the test sample, one need to compute the weights and reconstruction error and its complexity is $\mathcal{O}(mnpCD)$. Note that, the overall complexity of the proposed algorithms is the same as its special cases, even though more variables are introduced and computed.

4.8 Conclusion

We proposed joint sparsity and low-rank representation-based methods for multimodal recognition. The second formulation further enforce common representation across all the modality in order to get a more robust representation at the cost of losing information. Previously proposed joint sparsity or low-rank representation-based multimodal recognition methods are special cases of the proposed formulations. Efficient algorithms based on ADMM are derived to solve the proposed problems.

From the experimental results, we can conclude that: (1) for datasets, such as WVU dataset and Pascal-Sentence dataset, in which the performance of each modality is on the same level, there is no guarantee that enforcing a common representation (MRLRJS-C and RMRLRJS-C) may always yield better results because we get a more robust representation at the cost of losing information; (2) for datasets, such as the UMDAA dataset, in which the performance of each modality differs a lot, enforcing a common representation (MRLRJS and RMRLRJS) will successfully fuse all the modalities and perform much better than the general formulation (MRLRJS

and RMRLRJS) which fail to fuse strong and weak modalities together.

Chapter 5: Hierarchical Multimodal Metric Learning for Multimodal Classification

5.1 Introduction

Owing to recent developments in sensor technology, researchers and developers are able to collect multimodal data consisting of depth information and RGB images to achieve better performance for tasks such as object detection, classification and scene understanding [14, 15, 88–91]. Massive image and video data on Internet are associated with tags and metadata which are useful for image classification [92] and retrieval [93, 94]. Solutions to these problems can be formulated using multimodal classification frameworks. Multimodal classification has also been studied for other applications such as audio-visual speech classification [95, 96], and multimodal biometrics recognition [63, 64].

How to efficiently and effectively combine different modalities is the key issue in multimodal classification. Feature vectors corresponding to different modalities might be very different even if they essentially represent the same object. Some feature vectors are very discriminative while others are not; some feature vectors are clean while others are noisy; some feature vectors are dense while others are

sparse. Many factors like data acquisition, preprocessing and feature extraction can make feature vectors' behavior quite different. Therefore, direct linear combination of feature vectors or simple linear combination of the result of each modality can not guarantee good performance compared with using certain modality alone.

Metric learning algorithms can learn the Mahalanobis distance from data pairs and side information indicating the relationship of data pairs [3]. The learned distance metric can be better than Euclidean distance for the original feature space. Extensive research on metric learning in uni-modal setting is available in the literature. Classical algorithms includes the ones proposed in [3], Large Margin Nearest Neighbor (LMNN) algorithm [97] and Information Theoretical Metric Learning (ITML) algorithm [98]. When linear metric cannot adequately represent the inherent complexities that lie in the original feature space, various kernelized metric learning algorithms [99] [100] [101] [102] have been proposed to implicitly learn the metric in certain kernel space. For example, [102] demonstrated that a large class of Mahalanobis metric learning methods can be seen as learning a linear transformation (LT) kernel function and thus provided a constructive method for kernelizing these metric learning methods.

Extending the uni-modal metric learning algorithm to multi-modal metric learning can be a good solution for multimodal classification problems if the learned metrics are appropriate distance measures for corresponding feature spaces. Also, it is important to explore the relationship among the multiple metrics and the learning process should take into account the underlying differences among multiple modalities by balancing the contribution of each modality. As will be analyzed in Section

5.2 and Section 5.3, existing approaches for multimodal metric learning do not fully capture the relationships among the multiple learned ,metrics.

Motivated by previous works that consider shared representations in their formulations for multi-modal applications such as [64, 71, 95, 103, 104], we propose a Hierarchical Multimodal Metric Learning (HM3L) algorithm which fully explores the relationships among the different metrics of different modalities. In our formulation, metric of each modality is constructed through the multiplication of modality specific part representing appropriate subspace and a common part ($p.s.d$ matrix) shared by all the metrics. Figure 5.1 gives an overview of the proposed multimodal metric learning algorithm. Given multimodal representations, first we apply modality-specific projections $\mathbf{P}_k$ to each modality since their representations are very different in nature, then we apply the common metric $\mathbf{M}$ to features after the modality-specific projection assuming the features lie in the same common space. Furthermore, The kernelization of the proposed algorithm using the general kernel learning framework proposed in [102] leads to Kernelized Hierarchical Multimodal Metric Learning (KHM3L) algorithm.

The rest of this chapter is organized as follows. In Section 5.2, we review different metric learning algorithms. In Section 5.3, the Hierarchical Multimodal Metric Learning (HM3L) is proposed and differences from other related multiple metrics learning algorithms are discussed. In Section 5.4, the Kernelized Hierarchical Multimodal Metric Learning (KHM3L) is formulated as the nonlinear extension of the HM3L. In Section 5.5, efficient algorithms based on subgradient method are derived to solve the optimization problems corresponding to HM3L and KHM3L

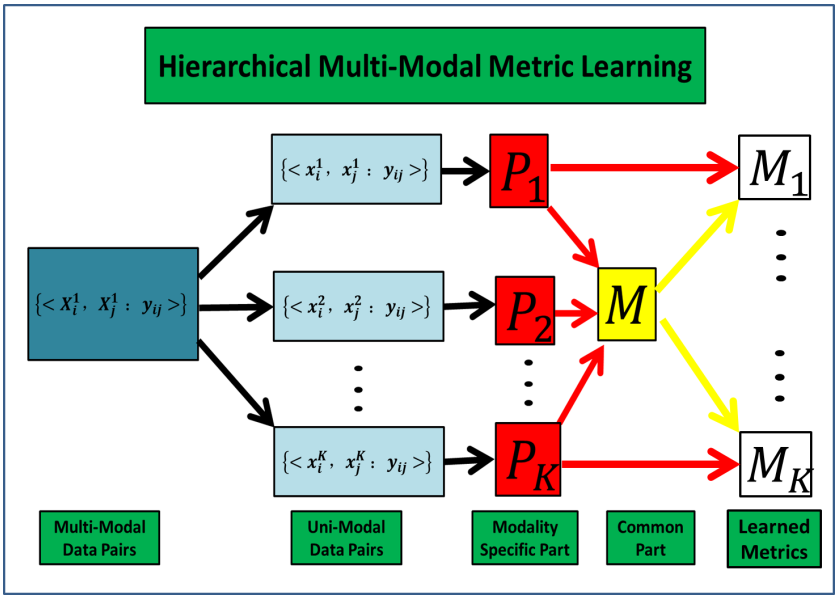


Figure 5.1: Overview of Hierarchical MultiModal Metric Learning.

respectively. Extensive experimental results on four datasets are presented in Section 5.6. Complexity analysis for HM3L and KHM3L is provided in Section 5.7. Finally, Section 5.8 concludes the paper with a brief summary.

5.2 Related Work

Metric learning has been studied in various fields such as machine learning [3, 97], information retrieval [105], computer vision [106] and biometrics [107, 108]. The goal of a metric learning algorithm is to learn a metric so that after data are projected using the learned metric, similar data samples (e.g. from the same class) are clustered together and dissimilar data samples (e.g. samples from different classes) are separated.

In a recent work, [3] formulated the metric learning problem as a convex opti-

mization problem by utilizing the side information of two data samples being similar or dissimilar. LMNN [97] applies the idea of large margin in Support Vector Machine (SVM) to improve the KNN classifier and uses triplet constraints to describe the relative relationships among three samples. In [98], the information theoretical metric learning (ITML) algorithm was proposed which essentially minimizes the differential relative entropy between two multivariate Gaussians under constraints on the distance function.

More recent metric learning algorithms also explore the structure of the metric by enforcing low-rank constraints [109, 110] or sparse constraints [111–113] or both sparse and low-rank constraints [114]. For high dimensional problems, [109] showed that enforcing low-rank constraints on the metric during the learning process is computationally efficient and tractable even with a small number of samples. More comprehensive survey of various metric learning methods and their applications are summarized in [115, 116].

Several multimodal metric learning algorithms have also been proposed in the literature. For instance, a multimodal metric learning method in [117] applies multi-wing harmonium (MWH) learning framework to get latent representations from different modalities and learns a metric under a probabilistic formulation. A Heterogeneous Multi-Metric Learning algorithm proposed in [118] for multi-sensor fusion essentially extends the LMNN algorithm [97] for multi-metric learning. Similarly, in [119] a large margin multi-metric learning (LM3L) was proposed for face and kinship verification which learns multiple metrics under which the correlations of different feature representations of each sample are maximized. Some of the other

multimodal metric learning algorithms include Pairwise-constrained Multiple Metric Learning (PMML) [120]. Note that these methods can be viewed as multimodal extensions of the classical unimodal metric learning algorithms like ITML and LMNN. One of the limitations of these methods is that they do not explore the relationships among different metrics corresponding to different modalities.

5.3 Formulation

5.3.1 Problem Description

Let

$$S = \{(X_i, X_j)|y_{ij} = 1\}$$

and

$$D = \{(X_i, X_j)|y_{ij} = -1\}$$

be two sets consisting of similar instance pairs and dissimilar instance pairs, respectively. An instance in the multimodal scenario is denoted as

$$X_i = \{x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(K)}\},$$

which consists of K features from K different modalities, where $x_i^{(1)} \in \mathbb{R}^{l_1}, x_i^{(2)} \in \mathbb{R}^{l_2}, \cdots, x_i^{(K)} \in \mathbb{R}^{l_K}$. Note that the dimension of each feature vector can be different. In multimodal metric learning, the objective is to learn metrics for such instances consisting of K feature vectors.

A simple way to learn a metric for multimodal data is by concatenating the features of the K modalities into one feature vector of length $\sum_{i=1}^{K} l_i$ and applying

the classical metric learning algorithms like LMNN or ITML. The drawback of this approach is the high computational cost incurred by learning an $\sum_{i=1}^{K} l_i$ by $\sum_{i=1}^{K} l_i$ metric. This problem is even more serious for high-dimensional multimodal data.

Existing multimodal metric leaning algorithms such as Pairwise-constrained Multiple Metric Learning [120], Large Margin Multi-metric Learning [119], and Heterogeneous Multi-Metric Learning [118], are extensions of the classical unimodal metric learning algorithms in which the distance between any two instances is obtained as

$$\begin{aligned} d_m^2(X_i, X_j) &= \frac{1}{K}\sum_{i=1}^{K} d_{M_k}^2(x_i^{(k)}, x_j^{(k)}) \\ &= \frac{1}{K}\sum_{i=1}^{K}(x_i^{(k)} - x_j^{(k)})^T \mathbf{M}_k (x_i^{(k)} - x_j^{(k)}). \end{aligned} \tag{5.1}$$

These approaches simultaneously solve K positive semi-definite $(p.s.d)$ matrices $\mathbf{M}_k, k = 1, \cdots, K$ as metrics in a joint formulation.

5.3.2 Hierarchical Multimodal Metric Learning (HM3L)

In order to efficiently learn multiple metrics for multiple modalities as well as to capture the relationship among them, we enforce the different metrics $\mathbf{M}_k, k = 1, \cdots, K$ to satisfy the following condition

$$\mathbf{M}_k = \mathbf{P}_k^T \mathbf{M} \mathbf{P}_k, \quad k = 1, \cdots, K, \tag{5.2}$$

where $\mathbf{P}_k \in \mathbb{R}^{d \times l_k}$ and $d \leq \min\{l_1, l_2, \cdots, l_K\}$. Also, $\mathbf{M}$ is required to be a $p.s.d$ matrix. Using this formulation, one can prove that if $\mathbf{M} \in \mathbb{R}^{d \times d}$ is $p.s.d$, then for any non-trivial $\mathbf{P}_k \in \mathbb{R}^{d \times l_k}$, $\mathbf{M}_k = \mathbf{P}_k^T \mathbf{M} \mathbf{P}_k$ is $p.s.d$.

For the given training data, the learned metrics $\mathbf{M}_k$ are obtained by learning modality specific part $\mathbf{P}_k$ and the shared part $\mathbf{M}$ in a hierarchical framework. As long as $\mathbf{M}$ is *p.s.d*, $\mathbf{M}_k$ is *p.s.d* meaning that $\mathbf{M}_k$ are valid metrics.

By enforcing (5.2), we establish the relationship among the different modalities. As a result, we can formulate the Hierarchical multimodal metric learning (HM3L) algorithm as the optimization problem specified in (5.3).

$$\begin{aligned} \min_{\mathbf{M}\in S_d^+} \quad & tr(\mathbf{M}) + \gamma \sum_{k=1}^{K} \|\mathbf{P}_i\|_F^2 \\ s.t. \ & \frac{1}{K}\sum_{k=1}^{K} d_M^2(\mathbf{P}_k x_i^{(k)}, \mathbf{P}_k x_j^{(k)}) \leq \mu \qquad \text{if } y_{ij} = 1 \\ & \frac{1}{K}\sum_{k=1}^{K} d_M^2(\mathbf{P}_k x_i^{(k)}, \mathbf{P}_k x_j^{(k)}) \geq \beta \qquad \text{if } y_{ij} = -1. \end{aligned} \tag{5.3}$$

Here γ controls the relative contribution to the cost function between $\mathbf{P}_k$ and $\mathbf{M}$ and μ and β are non-negative real numbers which specify the upper bound for distance of two similar instances and lower bound for distance of two dissimilar instances, respectively. We introduce the slack variables $\epsilon_{ij} > 0$ for constraints. Then (5.3) can be rewritten as

$$\begin{aligned} \min_{\mathbf{M}\in S_d^+} \quad & tr(\mathbf{M}) + \gamma \sum_{k=1}^{K} \|\mathbf{P}_i\|_F^2 \\ s.t. \ & \frac{1}{K}\sum_{k=1}^{K} d_M^2(\mathbf{P}_k x_i^{(k)}, \mathbf{P}_k x_j^{(k)}) \leq \mu + \epsilon_{ij} \qquad \text{if } y_{ij} = 1 \\ & \frac{1}{K}\sum_{k=1}^{K} d_M^2(\mathbf{P}_k x_i^{(k)}, \mathbf{P}_k x_j^{(k)}) \geq \beta - \epsilon_{ij} \qquad \text{if } y_{ij} = -1. \end{aligned} \tag{5.4}$$

5.3.3 HM3L-based multimodal classification

Once $\mathbf{P}_k$ and $\mathbf{M}$ are learned, we can easily get $\mathbf{L}$ such that $\mathbf{L}^T\mathbf{L} = \mathbf{M}$ through matrix decomposition. Then the multi-modal data

$$X_i = \{x_i^{(1)}, x_i^{(2)}, \cdots, x_i^{(K)}\}$$

can be projected by $\mathbf{P}_k$ and $\mathbf{L}$ and transformed to

$$\hat{X}_i = \{\mathbf{L}\mathbf{P}_1 x_i^{(1)}, \mathbf{L}\mathbf{P}_2 x_i^{(2)}, \cdots, \mathbf{L}\mathbf{P}_K x_i^{(K)}\}.$$

Concatenation of all the projected features can be used with various classification algorithms like KNN and SVM.

5.4 Kernelized Hierarchical Multimodal Metric Learning(KHM3L)

Very often a linear projection cannot capture the inherent complexities of given data. To address this limitation, various works introduce nonlinearity into the formulation by proposing kernelized metric learning algorithms in order to compute the Mahalanobis distance (linear projection) in some non-linear feature space.

Kernel function $\boldsymbol{\kappa} : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}$ is of the form $\boldsymbol{\kappa}(x, y) = \phi(x)^T\phi(y)$ for function ϕ which maps give instance x to some feature space $\mathcal{H}$. The dimensionality of feature space $\mathcal{H}$ is denoted as d_l and can be infinite. Some commonly used kernels include polynomial kernels

$$\boldsymbol{\kappa}(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle + a)^b$$

and Gaussian kernels

$$\boldsymbol{\kappa}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{c}\right)$$

where a, b and c are the parameters.

5.4.1 Kernelized metric learning for single-modal instances

The squared Mahalanobis distance of two instances in the $\mathcal{H}$ space can be denoted as:

$$d^2_{\mathbf{M}}(\phi(x_i), \phi(x_j)) = (\phi(x_i) - \phi(x_j))^T \mathbf{M} (\phi(x_i) - \phi(x_j)) \tag{5.5}$$

Where $\mathbf{M_k}$ is $p.s.d$ matrix in $\mathcal{H}$ space. Learning metric $\mathbf{M_k}$ in kernel space given finite pairs of instances being similar or dissimilar is an ill-posed problem since the dimensions of $\mathbf{M_k}$ can be infinite.

Kernelized metric learning does not explicitly learn $\mathbf{M}$. As proved in [102], for the following problem,

$$\begin{aligned} &\min_{\mathbf{M} \in S_d^+} \quad tr(\mathbf{M}) && && (5.6)\\ &s.t.\ d^2_M(\phi(x_i), \phi(x_j)) \leq \mu && \text{if } y_{ij} = 1 \\ &\quad\ d^2_M(\phi(x_i), \phi(x_j)) \geq \beta && \text{if } y_{ij} = -1 \end{aligned}$$

the optimal solution is of the form $\mathbf{M} = \boldsymbol{\Phi}(X)\mathbf{A}\boldsymbol{\Phi}(X)^T$ and $\mathbf{A}$ is a P.S.D matrix. $\mathbf{X}$ and $\boldsymbol{\Phi}(\mathbf{X})$ are defined as $[\mathbf{x}_1, \ldots, \mathbf{x}_N]$ and $[\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_N)]$ respectively assuming N training samples.

Therefore,(5.6) is equivalent to the following optimization problem,

$$\begin{aligned} \min_{\mathbf{A}\in S_d^+} \quad & tr(\mathbf{A}\boldsymbol{\mathcal{K}}) \\ s.t.\ & (\boldsymbol{\mathcal{K}_i}-\boldsymbol{\mathcal{K}_j})^T\mathbf{A}(\boldsymbol{\mathcal{K}_i}-\boldsymbol{\mathcal{K}_j}) \le \mu \qquad \text{if } y_{ij}=1 \\ & (\boldsymbol{\mathcal{K}_i}-\boldsymbol{\mathcal{K}_j})^T\mathbf{A}(\boldsymbol{\mathcal{K}_i}-\boldsymbol{\mathcal{K}_j}) \ge \beta \qquad \text{if } y_{ij}=-1 \end{aligned} \tag{5.7}$$

where $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{N\times N}$ is defined as $\boldsymbol{\Phi}(\mathbf{X})^T\boldsymbol{\Phi}(\mathbf{X})$ and it is a $p.s.d$ kernel matrix. $\boldsymbol{\mathcal{K}_{ij}} = \boldsymbol{\kappa}(x_i,x_j)$ and $\boldsymbol{\mathcal{K}_i} = [\boldsymbol{\kappa}(x_1,x_i),\cdots,\boldsymbol{\kappa}(x_N,x_i)]^T \in \mathbb{R}^{N\times 1}$. Note that the computation of $\boldsymbol{\mathcal{K}}$ only requires dot products without carrying out the mapping $\boldsymbol{\phi}$ and $\boldsymbol{\mathcal{K}}$ can be precomputed from the training data. This makes kernelized metric learning almost the same as linear metric learning.

5.4.2 Kernelized Hierarchical Multimodal Metric Learning

Corresponding to (5.5) for single-modal instances, the squared Mahalanobis distance of two multimodal instances in the kernel space can be denoted as:

$$\begin{aligned} d^2_{\mathbf{M}}(\phi(X_i),\phi(X_j)) &= \frac{1}{K}\sum_{i=1}^{K} d^2_{M_k}(\phi_k(x_i^{(k)}),\phi_k(x_j^{(k)})) \\ &= \frac{1}{K}\sum_{i=1}^{K}(\phi_k(x_i^{(k)})-\phi_k(x_j^{(k)}))^T\mathbf{M}_k(\phi_k(x_i^{(k)})-\phi_k(x_j^{(k)})). \end{aligned} \tag{5.8}$$

Let $\mathbf{M_k} = \boldsymbol{\Phi}_k(\mathbf{X}^{(k)})\mathbf{A_k}\boldsymbol{\Phi}_k(\mathbf{X}^{(k)})^T$ for $k=1,2,\cdots,K$. $\mathbf{A_k}$ is $(p.s.d)$ matrix for $k=1,\ldots,K$. $\mathbf{X}^{(k)}$ and $\boldsymbol{\Phi}_k(\mathbf{X}^{(k)})$ are defined as $[\mathbf{x}_1^{(k)},\cdots,\mathbf{x}_N^{(k)}]$ and $[\phi_k(\mathbf{x}_1^{(k)}),\cdots,\phi_k(\mathbf{x}_N^{(k)})]$ respectively assuming N training samples for the kth modality. Note that here the training samples may or may not have labels in real-settings and $\mathbf{X}^{(k)}(k=1,\ldots,K)$ can be constructed by concatenation of the data samples from given similar pairs and dissimilar pairs.

In order to efficiently learn multiple metrics for multiple modalities and capture the relationships among them, we enforce $\mathbf{A}_k, k = 1, \cdots, K$ to satisfy the following condition

$$\mathbf{A}_k = \mathbf{P}_k^T \mathbf{M} \mathbf{P}_k, \quad k = 1, \cdots, K, \tag{5.9}$$

where $\mathbf{P}_k \in \mathbb{R}^{d \times N}$ and $d \leq N$. Also, $\mathbf{M}$ is required to be a $p.s.d$ matrix. Therefore, metrics in the kernel space for the K modalities satisfy

$$\mathbf{M}_k = \mathbf{\Phi}_K(\mathbf{X}^{(k)}) \mathbf{P}_k^T \mathbf{M} \mathbf{P}_k \mathbf{\Phi}_K(\mathbf{X}^{(k)})^T, \quad k = 1, \cdots, K, \tag{5.10}$$

By enforcing (5.10), we establish the relationships among the different modalities. As a result, we can formulate the Kernelized Hierarchical multimodal metric learning (KHM3L) algorithm as the following optimization problem with slack variables $\epsilon_{ij} > 0$ introduced for constraints,

$$\begin{aligned} \min_{\mathbf{M} \in S_d^+} \quad & tr(\mathbf{M}) + \gamma \sum_{k=1}^{K} tr(\mathbf{P}_k \boldsymbol{\mathcal{K}}^{(k)} \mathbf{P}_k^T) \\ s.t. \ & \frac{1}{K} \sum_{k=1}^{K} d_M^2(\mathbf{P}_k \boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_k \boldsymbol{\mathcal{K}_j}^{(k)}) \leq \mu + \epsilon_{ij} \qquad \text{if } y_{ij} = 1 \\ & \frac{1}{K} \sum_{k=1}^{K} d_M^2(\mathbf{P}_k \boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_k \boldsymbol{\mathcal{K}_j}^{(k)}) \geq \beta - \epsilon_{ij} \qquad \text{if } y_{ij} = -1. \end{aligned} \tag{5.11}$$

where $\boldsymbol{\mathcal{K}}^{(k)}$ is defined as $\mathbf{\Phi}_k(\mathbf{X}^{(k)})^T \mathbf{\Phi}_k(\mathbf{X}^{(k)})$ for the kth modality and it is a $p.s.d$ kernel matrix. $\boldsymbol{\mathcal{K}_{ij}}^{(k)} = \boldsymbol{\kappa}_k(x_i^{(k)}, x_j^{(k)})$ and $\boldsymbol{\mathcal{K}_i}^{(k)} = [\boldsymbol{\kappa}_k(x_1^{(k)}, x_i^{(k)}), \cdots, \boldsymbol{\kappa}_k(x_N^{(k)}, x_i^{(k)})]^T \in \mathbb{R}^{N \times 1}$.

Here γ controls the relative contribution to the cost function between $\mathbf{P}_k \boldsymbol{\mathcal{K}}^{(k)} \mathbf{P}_k^T$ and $\mathbf{M}$. μ and β are non-negative real numbers which specify the upper bound for distance of two similar instances and lower bound for distance of two dissimilar instances, respectively.

5.4.3 KHM3L-based multimodal classification

Once $\mathbf{P}_k$ and $\mathbf{M}$ are learned, we can easily get $\mathbf{L}$ such that $\mathbf{L}^T\mathbf{L} = \mathbf{M}$ through matrix decomposition. Then the multi-modal data

$$X_i = \{x_i^{(1)}, \cdots, x_i^{(K)}\}$$

can be transformed to

$$\hat{X}_i = \{\mathbf{L}\mathbf{P}_1\boldsymbol{\mathcal{K}_i}^{(1)}, \mathbf{L}\mathbf{P}_2\boldsymbol{\mathcal{K}_i}^{(2)}, \cdots, \mathbf{L}\mathbf{P}_K\boldsymbol{\mathcal{K}_i}^{(K)}\}.$$

Concatenation of all the projected features can be used with various classification algorithms like KNN and SVM.

5.5 Optimization

5.5.1 Optimization for HM3L

To solve the proposed optimization problem (5.4), we apply hinge-loss function to get rid of the constraints which results in an unconstrained optimization problem as follows

$$\begin{aligned}
&\min_{\mathbf{M}\in S_d^+} tr(\mathbf{M}) + \gamma \sum_{k=1}^{K} \|\mathbf{P}_i\|_F^2 \\
&+ \alpha C \sum_{(X_i,X_j)\in S} \left[\frac{1}{K} \sum_{k=1}^{K} d_M^2(\mathbf{P}_k x_i^{(k)}, \mathbf{P}_k x_j^{(k)}) - \mu \right]_+ \\
&+ (1-\alpha) C \sum_{(X_i,X_j)\in D} \left[\beta - \frac{1}{K} \sum_{k=1}^{K} d_M^2(\mathbf{P}_k x_i^{(k)}, \mathbf{P}_k x_j^{(k)}) \right]_+
\end{aligned} \tag{5.12}$$

where C is a positive number that controls the relative contribution between the constraints on the metric and the constraints on the data samples, α is a constant

that balances the relative contribution between the pairs from similar set and pairs from dissimilar set. Let $L(\mathbf{M}; \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_K)$ denote the above cost function we are trying to minimize. It is a bi-convex optimization problem when we consider $\mathbf{P}_k$ $(k = 1, 2, ..., K)$ together as $\mathbf{P}$. We iteratively solve for $\mathbf{M}$ and $\mathbf{P}$ by updating one with the other fixed.

The hinge-loss function indicates that only pairs of samples that violate the distance constraints will make contributions to the overall cost function. For notational convenience, let $A^t_{S,P}$, $A^t_{D,P}$, $A^t_{S,M}$ and $A^t_{D,M}$ denote active sets at time t. $A^t_{S,P}$ ($A^t_{D,P}$) means set for similar (dissimilar) pairs that violate the distance constraint when we fix $\mathbf{P}_k$ to update $\mathbf{M}$. Similarly, $A^t_{S,M}$ ($A^t_{D,M}$) means set for similar (dissimilar) pairs that violate the distance constraint when we fix $\mathbf{M}$ to update $\mathbf{P}_k$.

$$A^t_{S,P} = \{(X_i, X_j) \in S | \frac{1}{K} \sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1} x_i^{(k)}, \mathbf{P}_{k,t-1} x_j^{(k)}) \geq \mu\}$$

$$A^t_{D,P} = \{(X_i, X_j) \in D | \frac{1}{K} \sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1} x_i^{(k)}, \mathbf{P}_{k,t-1} x_j^{(k)}) \leq \beta\}$$

$$A^t_{S,M} = \{(X_i, X_j) \in S | \frac{1}{K} \sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1} x_i^{(k)}, \mathbf{P}_{k,t-1} x_j^{(k)}) \geq \mu\}$$

$$A^t_{D,M} = \{(X_i, X_j) \in D | \frac{1}{K} \sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1} x_i^{(k)}, \mathbf{P}_{k,t-1} x_j^{(k)}) \leq \beta\}.$$

Updating $\mathbf{M}$

Fixing $\mathbf{P}_k$, projected sub-gradient method [121] can be applied to solve for $\mathbf{M}$. It involves two key steps.

Step 1:

$$\mathbf{M}_{tmp} = \mathbf{M}_t - \eta g_t(\mathbf{M}), \tag{5.13}$$

where $g_t(\mathbf{M})$ is the gradient of $L(\mathbf{M})$ at time t and it is derived as,

$$g_t(\mathbf{M}) = \mathbf{I}_{d\times d} + C\alpha \sum_{(X_i,X_j)\in A^t_{S,P}} \left[\frac{1}{K}\sum_{k=1}^{K} \mathbf{P}_{k,t-1} B^{(k)}_{i,j} \mathbf{P}^T_{k,t-1}\right] + C(1-\alpha) \sum_{(X_i,X_j)\in A^t_{D,P}} \left[-\frac{1}{K}\sum_{k=1}^{K} \mathbf{P}_{k,t-1} B^{(k)}_{i,j} \mathbf{P}^T_{k,t-1}\right] \tag{5.14}$$

Where $B^{(k)}_{i,j} = (x^{(k)}_i - x^{(k)}_j)(x^{(k)}_i - x^{(k)}_j)^T$ is a rank 1 matrix.

Step 2:

$$\mathbf{M}_{t+1} = \mathbf{V}^T[\mathbf{\Sigma}]_+\mathbf{V}, \tag{5.15}$$

where $\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}$ is the eigenvalue decomposition of $\mathbf{M}_{tmp}$. Projecting $\mathbf{M}_{tmp}$ onto the $p.s.d$ cone can be done by thresholding the eigenvalues by keeping the positive eigenvalues and setting the negative ones to be 0.

Updating $\mathbf{P}$

Fixing $\mathbf{M}$, each $\mathbf{P}_k$ can be updated separately through gradient descent as

$$\mathbf{P}_{k,t} = \mathbf{P}_{k,t-1} - \eta g_t(\mathbf{P}_k), \quad k = 1, 2, ..., K, \tag{5.16}$$

where $g_t(\mathbf{P}_k)$ is the gradient of $L(\mathbf{P}_k)$ at time t and it is derived as

$$g_t(\mathbf{P}_k) = 2\gamma\mathbf{P}_{k,t-1} + C\alpha \sum_{(X_i,X_j)\in A^t_{S,M}} \left[\frac{2}{K}\mathbf{M}_t\mathbf{P}_{k,t-1}B^{(k)}_{i,j}\right] + C(1-\alpha) \sum_{(X_i,X_j)\in A^t_{D,M}} \left[-\frac{2}{K}\mathbf{M}_t\mathbf{P}_{k,t-1}B^{(k)}_{i,j}\right] \tag{5.17}$$

The overall Hierarchical Multimodal Metric Learning (HM3L) algorithm is summarized in Algorithm 1.

Algorithm 4: Hierarchical Multimodal Metric Learning (HM3L)

Inputs:

$S = \{(X_i, X_j)|y_{ij} = 1\}$, $D = \{(X_i, X_j)|y_{ij} = -1\}$, positive integer γ, α, η, μ, β, C and maximum iteration T.

Initialization:

To initialize $\mathbf{P}_k$ (k = 1,2,...,K):

construct $\mathbf{X}^k \in \mathbb{R}^{l_k \times N}$ of $x_i^{(k)}$ from S and D;

perform PCA on $\mathbf{X}^k$ to obtain $\mathbf{P}_{k,0} \in \mathbb{R}^{d \times l_k}$.

To initialize $\mathbf{M}$:

set $\mathbf{M}_0 = \mathbf{I}_{d \times d}$.

Main loop:

for $t = 1 : T$ **do**

calculate $A_{S,P}^t$ and $A_{D,P}^t$ to update $\mathbf{M}$ through (5.14), (5.13) and (5.15);

calculate $A_{S,M}^t$ and $A_{D,M}^t$ to update $\mathbf{P}_k$ through (5.17) and (5.16).

end

Outputs:

$\mathbf{P}_k$ $(k = 1, 2, \ldots, K)$ and $\mathbf{M}$.

5.5.2 Optimization for KHM3L

First, we transform the original multimodal instance pairs. Given similar (dissimilar) multimodal instance pairs set S (D), construct $\mathbf{X}^k \in \mathbb{R}^{l_k \times N}$ of $x_i^{(k)}$ from S and D (for $k = 1, \ldots, K$) and compute the kernel matrix $\boldsymbol{\mathcal{K}}^{(k)} = \boldsymbol{\Phi}_k(\mathbf{X}^{(k)})^T \boldsymbol{\Phi}_k(\mathbf{X}^{(k)})$ using the kernel function $\boldsymbol{\kappa}_k$. Then, redefine

$$S = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) | y_{ij} = 1\}$$

and

$$D = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) | y_{ij} = -1\}$$

where $\boldsymbol{\mathcal{K}}_i = \{\boldsymbol{\mathcal{K}}_i^{(1)}, \cdots, \boldsymbol{\mathcal{K}}_i^{(K)}\}$.

The optimization of KHM3L follows similar steps for that of HM3L. The optimization problem denoted by (5.11) is equivalent to the following unconstrained optimization problem

$$\begin{aligned} &\min_{\mathbf{M} \in S_d^+} tr(\mathbf{M}) + \gamma \sum_{k=1}^{K} tr(\mathbf{P}_k \boldsymbol{\mathcal{K}}^{(k)} \mathbf{P}_k^T) \\ &+ \alpha C \sum_{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) \in S} \left[\frac{1}{K} \sum_{k=1}^{K} d_M^2(\mathbf{P}_k \boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_k \boldsymbol{\mathcal{K}_j}^{(k)}) - \mu \right]_+ \\ &+ (1-\alpha) C \sum_{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) \in D} \left[\beta - \frac{1}{K} \sum_{k=1}^{K} d_M^2(\mathbf{P}_k \boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_k \boldsymbol{\mathcal{K}_j}^{(k)}) \right]_+ \end{aligned} \tag{5.18}$$

where the hyperparameters C and α are the same as in (5.12). Let $L(\mathbf{M}; \mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_K)$ denote the above cost function and as this is still a bi-convex optimization problem, we iteratively solve for $\mathbf{M}$ and $\mathbf{P}$ by updating one with the other fixed.

Active sets $A^t_{S,P}$, $A^t_{D,P}$, $A^t_{S,M}$ and $A^t_{D,M}$ at iteration t are defined as

$$A^t_{S,P} = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) \in S | \frac{1}{K}\sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_j}^{(k)}) \geq \mu\}$$

$$A^t_{D,P} = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) \in D | \frac{1}{K}\sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_j}^{(k)}) \leq \beta\}$$

$$A^t_{S,M} = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) \in S | \frac{1}{K}\sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_j}^{(k)}) \geq \mu\}$$

$$A^t_{D,M} = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j) \in D | \frac{1}{K}\sum_{k=1}^{K} d^2_{\mathbf{M}_{t-1}}(\mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_i}^{(k)}, \mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}_j}^{(k)}) \leq \beta\}.$$

Updating $\mathbf{M}$

Updating $\mathbf{M}$ requires the same two steps as specified by (5.13) and (5.15). $g_t(\mathbf{M})$ is derived as,

$$g_t(\mathbf{M}) = \mathbf{I}_{d\times d} + C\alpha \sum_{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j)\in A^t_{S,P}} \left[\frac{1}{K}\sum_{k=1}^{K} \mathbf{P}_{k,t-1} B^{(k)}_{i,j} \mathbf{P}^T_{k,t-1}\right] + C(1-\alpha) \sum_{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j)\in A^t_{D,P}} \left[-\frac{1}{K}\sum_{k=1}^{K} \mathbf{P}_{k,t-1} B^{(k)}_{i,j} \mathbf{P}^T_{k,t-1}\right] \tag{5.19}$$

Where $B^{(k)}_{i,j} = (\boldsymbol{\mathcal{K}_i}^{(k)} - \boldsymbol{\mathcal{K}_j}^{(k)})(\boldsymbol{\mathcal{K}_i}^{(k)} - \boldsymbol{\mathcal{K}_j}^{(k)})^T$.

Updating $\mathbf{P}$

Fixing $\mathbf{M}$, each $\mathbf{P}_k$ can be updated separately through gradient descent as

$$\mathbf{P}_{k,t} = \mathbf{P}_{k,t-1} - \eta g_t(\mathbf{P}_k), \quad k = 1, 2, ..., K, \tag{5.20}$$

where $g_t(\mathbf{P}_k)$ is the gradient of $L(\mathbf{P}_k)$ at time t and it is derived as

$$g_t(\mathbf{P}_k) = 2\gamma \mathbf{P}_{k,t-1}\boldsymbol{\mathcal{K}}^{(k)} + C\alpha \sum_{(\boldsymbol{\mathcal{K}}_i,\boldsymbol{\mathcal{K}}_j)\in A^t_{S,M}} \left[\frac{2}{K}\mathbf{M}_t\mathbf{P}_{k,t-1}B^{(k)}_{i,j}\right] + C(1-\alpha) \sum_{(\boldsymbol{\mathcal{K}}_i,\boldsymbol{\mathcal{K}}_j)\in A^t_{D,M}} \left[-\frac{2}{K}\mathbf{M}_t\mathbf{P}_{k,t-1}B^{(k)}_{i,j}\right] \tag{5.21}$$

The overall Kernelized Hierarchical Multimodal Metric Learning (KHM3L) algorithm is summarized in Algorithm 2.

5.6 Experiments

To illustrate the effectiveness of our method, we present experimental results on four publicly available multimodal datasets: RGB-D Object dataset [14], CIN 2D3D object dataset [88], SUN RGB-D dataset [15] and NUS-WIDE dataset [122]. The details of these datasets, experimental setups and experimental results are given in the following subsections.

For experiments on each dataset, we include (1) the baseline result (without metric learning) obtained by certain features plus either NN or SVM classifiers depending on which was used to report the baseline result, (2) the proposed HM3L method as well as other publicly available multiple metrics learning methods [118,120] to first transform the features used in the baseline result, then apply NN or SVM classifier, (3) other methods which reported the best results on that experiment.

5.6.1 Object recognition on RGB-D Object dataset

RGB-D Object dataset [14] is a large scale multi-view dataset for 3D object recognition, segmentation, scene labeling and so on. It consists of video recordings of 300

Algorithm 5: Kernelized Hierarchical Multimodal Metric Learning (KHM3L)

Inputs:

$S = \{(X_i, X_j)|y_{ij} = 1\}$, $D = \{(X_i, X_j)|y_{ij} = -1\}$, kernel functions $\boldsymbol{\kappa}_k$ with their associated parameters $(k = 1, \ldots, K)$, positive real numbers γ, α, η, μ, β, C and maximum iteration T.

Preprocessing:

To transform multimodal instance (k= 1,2,...,K):

construct $\mathbf{X}^k \in \mathbb{R}^{l_k \times N}$ of $x_i^{(k)}$ from S and D ;

compute kernel matrix: $\boldsymbol{\mathcal{K}}^{(k)} = \boldsymbol{\Phi}(\mathbf{X}^{(k)})^T \boldsymbol{\Phi}(\mathbf{X}^{(k)})$.

Redefine $S = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j)|y_{ij} = 1\}$;

Redefine $D = \{(\boldsymbol{\mathcal{K}}_i, \boldsymbol{\mathcal{K}}_j)|y_{ij} = -1\}$.

Initialization:

To initialize $\mathbf{P}_k$ (k = 1,2,...,K):

perform PCA on $\boldsymbol{\mathcal{K}}^{(k)}$ to obtain $\mathbf{P}_{k,0} \in \mathbb{R}^{d \times N}$.

To initialize $\mathbf{M}$:

set $\mathbf{M}_0 = \mathbf{I}_{d \times d}$.

Main loop:

for $t = 1 : T$ **do**

calculate $A_{S,P}^t$ and $A_{D,P}^t$ to update $\mathbf{M}$ through (5.19), (5.13) and (5.15);

calculate $A_{S,M}^t$ and $A_{D,M}^t$ to update $\mathbf{P}_k$ through (5.21) and (5.20).

end

Outputs:

$\mathbf{P}_k$ $(k = 1, 2, \ldots, K)$ and $\mathbf{M}$.

everyday objects organized into 51 different categories. The video recordings were captured by cameras mounted at 3 different elevation angles of 30^0, 45^0 and 60^0. A single RGB-D frame consists of both an RGB image and a depth image. Evaluation protocol for various computer vision tasks such as instance recognition and category recognition were set in [14]. RGB-D Images were sampled every 5th frame of the videos and in total about 45,000 RGB-D images were collected.

Kernel descriptors [123] [124] were extracted as features for RGB images and depth image. For RGB images, the LBP kernel descriptor, Gradient kernel descriptor and normalized color kernel descriptor were extracted. For depth images, the gradient kernel descriptor and the LBP kernel descriptor were extracted from depth images; normal kernel descriptor and size kernel descriptor were extracted from point clouds which were converted from the depth images. For each kernel descriptor, object-level features were obtained from 1000 dimensional basis vector for 1×1, 2×2, 3×3 pyramid sub-regions. The basis vector was learned by K-means on about 400,000 sample kernel descriptors from training data. The dimensionality of each kernel descriptor is $(1 + 4 + 9) \times 1000 = 14000$; principal component analysis was used to reduce dimensionality to 1000. After feature extraction, each RGB-D image was represented by seven kernel descriptors and each kernel descriptor is 1000 dimensional vector.

Experimental Setup

For the instance recognition experiment, images corresponding to videos captured at angles 30^0 and 60^0 were used for training, and images corresponding to videos captured at angle 45^0 were used for testing. For the category recognition experiment, one object was randomly chosen and left out from each category for testing and all views of the remaining

objects were used for training. Ten trials were repeated for category recognition.

For the instance and category recognition tasks, we first learn multiple metrics for seven kernel descriptors using the similar and dissimilar set of the RGB-D images generated from the training data. We then perform linear SVM classification [125] based on the learned metrics. We also compare the performance of our method with the results reported in [103] which are based on deep learning-based methods for RGB-D image classification.

Methods	RGB	Depth	RGB-D
Lai [14]	60.7	46.2	74.8
Bo [124]	90.8	54.7	91.2
Blum [126]	82.9	-	90.4
HMP [127]	92.1	51.7	92.8
MMSS [103]	-	-	94.0
PMML [120] + linear SVM	92.7	53.4	92.9
HMML [118] + linear SVM	90.0	51.9	92.1
HM3L + linear SVM	**93.34**	**55.6**	**95.0**

Table 5.1: Instance recognition accuracy on RGB-D Object dataset.

Experiment Results

Classification results for instance recognition and category recognition are shown in Table 5.1 and Table 5.2 respectively. From these tables, we made the following observations. (1) the proposed HM3L-based classification method outperform the best results obtained from MMSS [103] which applies deep architectures on the RGB-D images for both instance recognition testing on over 13800 instances and category recognition over-

Methods	RGB	Depth	RGB-D
Lai [14]	64.7±2.2	74.5±3.1	83.8 ± 3.5
Bo [124]	80.7±2.1	80.3±2.9	86.5 ±2.1
Blum [126]	-	-	86.4 ±2.3
HMP [127]	**82.4 ± 3.1**	**81.2 ± 2.3**	87.5 ±2.9
MMSS [103]	-	-	88.5 ± 2.2
PMML [120] + linear SVM	80.2	77.7 ± 2.4	88.5 ± 1.4
HMML [118] + linear SVM	75.8± 3.2	77.4 ± 2.4	87.3 ± 1.8
HM3L + linear SVM	81.0 ± 2.7	79.1 ± 2.4	**89.2 ± 1.6**

Table 5.2: Category recognition accuracy on RGB-D Object dataset.

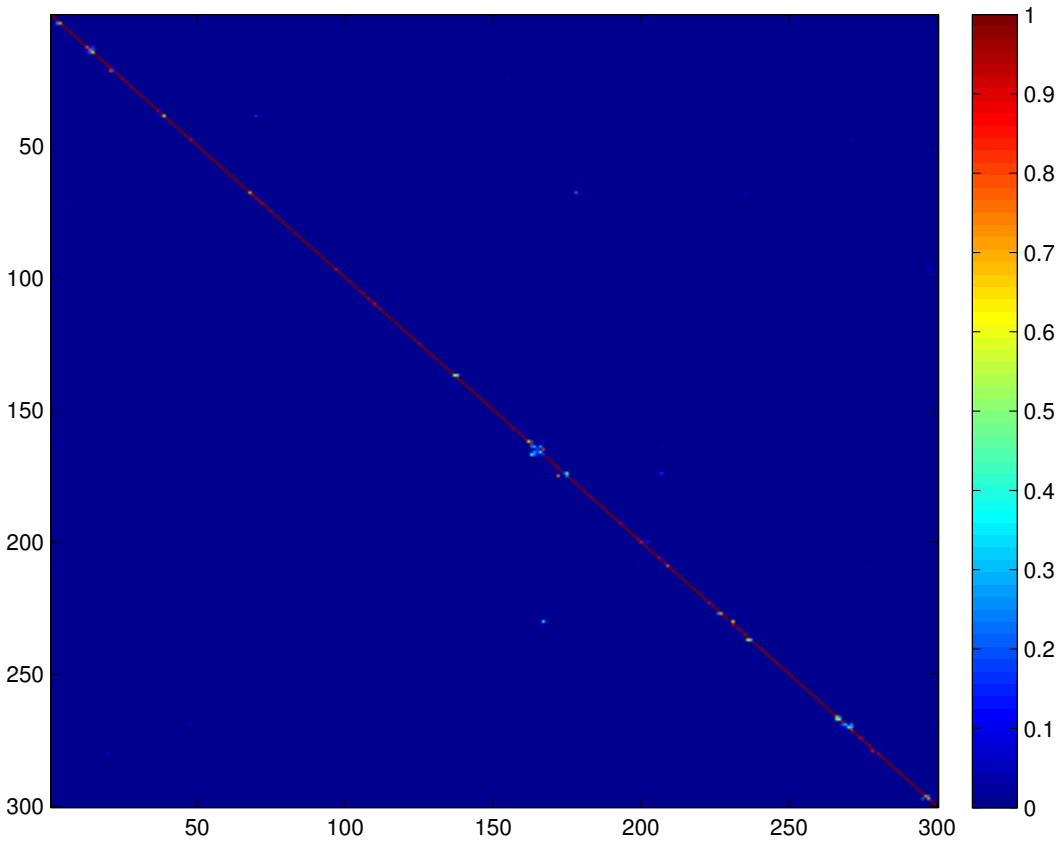


Figure 5.2: Confusion matrix for Instance recognition result.

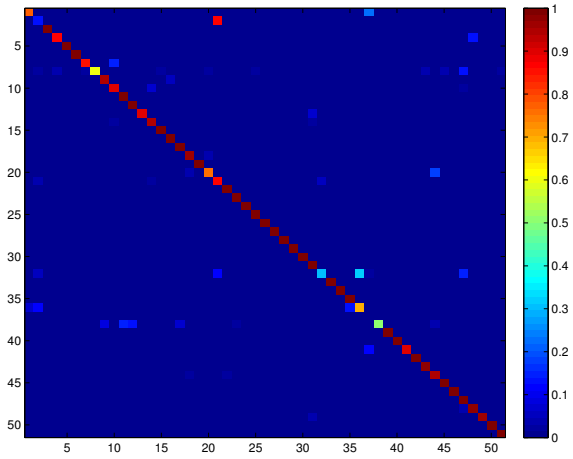


Figure 5.3: Confusion matrix for 8th trial category recognition result.

Figure 5.4: Examples of prediction errors in category recognition experiment.

all ten trials. (2) The proposed HM3L algorithm can boost the classification accuracy compared to the case where metrics learning was not performed. (3) HM3L-based multi-modal classification outperforms other multiple metrics learning-based classification and this shows that the idea of capturing the relationship for different multiple metrics can help to learn more appropriate distance measures.

Confusion matrices of classification results based on the proposed algorithm are shown in Figure 5.2 for instance recognition experiment and in Figure 5.3 for the 8th trial of category recognition experiment. The testing data of recognition experiment are placed such that testing samples of the same objects are put together and objects from

the same category are grouped together. As we can see from Figure 5.2, for each of 300 objects, most samples are classified correctly (diagonal) and many errors are made due to the misclassification of certain samples to other objects from the same category. Examples of misclassification in category recognition is shown in Figure 5.4. For each column, the objects on top was misclassified to the category represented by certain object in the bottom. We can see that errors occur due to similar color and shape.

5.6.2 Object recognition on CIN 2D3D dataset

CIN 2D3D object classification dataset [88] contains segmented color and depth images of 154 objects from eighteen categories of common household and office objects. Each category contains between three to fourteen objects. Each object was recorded using a high-resolution color camera and a time-of-flight rang sensor. Objects were rotated using a turn table and snapshots taken every ten degrees and yields 36 views per object. Each view is one data sample consisting of RGB image and Depth image. Following the similar procedures used to extract kernel descriptors for samples in RGB-D object dataset, we also extract kernel descriptors for data samples in 2D3D dataset.

Experiment Results

The evaluation protocol for category classification was set in the original paper [88]. six objects per category were used for training and remaining objects were used for testing. For each object, eighteen views are selected for training and eighteen views for testing. The training set consisted of 82 objects with a total of 1476 views. The test set contained 74 objects with 1332 views. Same methods as included in RGB-D dataset are evaluated. Classification results for category recognition are shown in Table 5.3. As can be seen from

this table, the proposed HM3L-based multimodal classification gives the best performance on average.

Methods	RGB	Depth	RGB-D
Browatzki [88]	66.6	74.6	82.8
HMP [127]	86.3	**87.6**	91.0
MMSS [103]	-	-	91.3
PMML [120] + linear SVM	**90.6**	82.7	91.8
HMML [118] + linear SVM	86.8	83.4	90.8
HM3L + linear SVM	89.9	86.4	**92.9**

Table 5.3: Category recognition accuracy (in %) on CIN 2D3D dataset.

5.6.3 Scene Categorization on SUN RGB-D dataset

SUN RGB-D dataset [15] consists of 10355 RGB-D scene images including 3784 Kinect v2 images, 1159 Intel RealSense images as well as 1449 images taken from the NYU Depth Dataset V2 [90], 554 scene images from the Berkeley B3DO Dataset [89], and 3389 Asus Xtion images from SUN3D videos [91]. We choose the same Places-CNN [128] scene features of dimension 4096 for both RGB image and depth image which were used to report the baseline results in [15].

Experimental Results

We followed the standard experimental setup for scene categorization task according to [15]. Specifically, nineteen scene categories with more than eighty images are used. These scene categories are bathroom, bedroom, classroom, computer room, conference

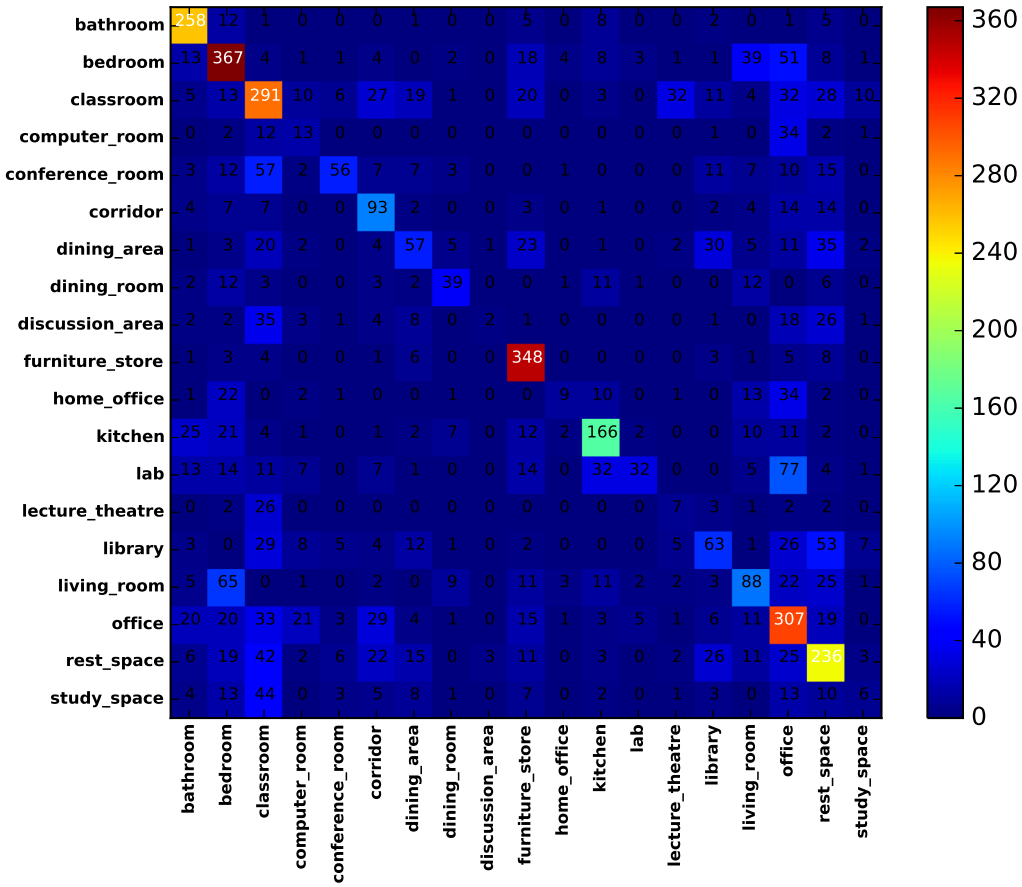


Figure 5.5: Confusion matrix for scene recognition result.

room, corridor, dining area, dining room, discussion area, furniture store, home office, kitchen, lab, lecture theatre, library, living room, office, rest space, study space.

The train and test split is available in [15]. In total, 4845 samples are used for training and 4659 samples are used for testing. The standard average categorization accuracy is used for evaluation. We apply the proposed HM3L method to the Places-CNN features, transform the original features with the learned matrices, and then apply one-vs-all rbf SVM for classification. The scene category recognition results are shown in Table 5.4 and the confusion matrix of scene recognition results based on the proposed algorithm is shown in Figure 5.5.

From results, we made the following observations. (1) the proposed HM3L-based classification method outperformed the best results obtained from [129, 130]. (2) The proposed HM3L algorithm as well other two multiple metrics learning algorithms can significantly boost the classification accuracy compared to the baseline case in which metrics learning was not performed. (3) HM3L-based multimodal classification outperforms other multiple metrics learning-based classification and this again shows the importance of capturing the relationship for different multiple metrics in the learning process.

5.6.4 Tagged image classification on NUS-WIDE dataset

The NUS-WIDE dataset [122] consists of 269,648 web images and tags from Flickr. For a fair comparison with previous results reported in [117], same subset of tagged images, same train/test splitting, same sets of similar (dissimilar) pairs of instances and same feature extraction procedures are applied. A subset of 1521 tagged images are used. These tagged images consist of 30 classes (actor, airplane, bicycle, bridge, buddha, building, butterfly, camels, car, cathedral, cliff, clouds, coast, computers, desert, flag, flowers,

Methods	RGB	Depth	RGB-D
Place-CNN + linear SVM [15]	35.6	25.5	37.2
Place-CNN + rbf SVM [15]	38.1	27.7	39.0
Liao [131]	36.1	-	41.3
Zhu [130]	-	-	41.5
Wang [129]	-	-	48.1
PMML [120] + rbf SVM	40.7	30.5	44.2
HMML [118] + rbf SVM	47.9	32.6	51.1
HM3L + rbf SVM	**48.6**	**33.2**	**52.3**

Table 5.4: Scene categorization accuracy (in %) on SUN RGB-D dataset.

food, forest, glacier, hills, lake, leaf, monks, moon, motorcycle, mushrooms, ocean, police, pyramid) and roughly fifty tagged images per class are randomly selected. By randomly splitting the dataset, 765 tagged images are used as training data and the remaining are used as testing data. From the training data, 9613 pairs of similar instances and 10067 pairs of dissimilar instances are selected to learn distance metrics. For images, 1024-D bag of visual words based on SIFT descriptors is extracted to represent the image modality; for tags, 1000-D bag of words is extracted to represent the associated tag modality. Therefore, one instance of tagged image is represented by feature vectors of two modalities.

Experiment Setup

For every approach considered, metrics are first learned. Then, KNN classification under the learned metrics is performed using training and testing data. The value of K is chosen to be 1, 3, 5, 10 and 20. We compare the performance of our method with those of "Xing + Original", "ITML+Original", "Xing + MWH", "ITML + MWH", "MKE" [132],

Methods	Xing+Original	ITML+Original	Xing+MWH	ITML+MWH	MKE [132]	Xie [117]	PMML [120]	HMML [118]	HM3L
1-NN	0.8995	0.8995	0.8995	0.9286	0.8056	0.9352	0.9233	0.9140	**0.9524**
3-NN	0.8108	0.6653	0.8849	0.8929	0.6944	0.9021	0.9220	0.9246	**0.9431**
5-NN	0.6971	0.4868	0.8426	0.8519	0.5860	0.8849	0.9299	0.9114	**0.9418**
10-NN	0.4775	0.2394	0.7646	0.7394	0.4405	0.8333	0.9139	0.9008	**0.9339**
20-NN	0.1548	0.0450	0.6230	0.4841	0.1746	0.7130	0.9074	0.8876	**0.9223**

Table 5.5: KNN Classification Accuracy under learned metrics for tagged images.

Heterogeneous Multi-Metric Learning (HMML) [118] and PMML [120]. "Xing+Original" and "ITML+Original" methods essentially apply algorithms proposed in [3] and [98] on the concatenated feature vectors from different modalities. Similarly, "Xing+MWH" and "ITML+MWH" correspond to the algorithms combined with the MWH model proposed in [117]. All parameters are tuned using cross-validation on training data.

Experimental Results

Table 5.5 shows the KNN classification accuracies of different methods. As can be seen from the table, the proposed method performed the best. This experiment clearly shows that our method can provide better distance measures which can enhance the performance of a classification algorithm.

To see whether the proposed algorithms converge, we empirically show the convergence of our algorithm by plotting the normalized cost function values versus iterations. From Figure 5.6, we can observe that the proposed algorithm converges in a few iterations.

5.7 Complexity Analysis

To analyze the computational complexity of the proposed methods, each step involves various matrices operation. In general, the complexity of matrix multiplication for

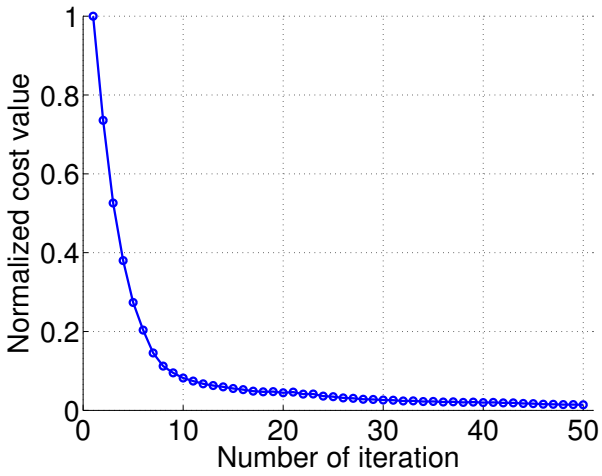


Figure 5.6: Normalized cost function over iterations.

a $m \times n$ matrix and a $n \times p$ matrix is $\mathcal{O}(mnp)$ and the complexity of matrix addition is $\mathcal{O}(mn)$ for two $m \times n$ matrices. The complexity of matrix inversion and singular value decomposition is $\mathcal{O}(n^3)$ for an $n \times n$ matrix.

For simplicity, Let's assume the number of modalities is K; the dimension of matrix $\mathbf{M}$ is $d \times d$; the dimension of the feature vector corresponding to different modality is l and thus the dimension of matrix $\mathbf{P}_k$ $(k = 1, \ldots, K)$ is $d \times l$; the number of training samples is N; set S consists of N_S similar pairs; set D consists of N_D dissimilar pairs; the number of iterations is T.

For the proposed HM3L algorithm, the complexity of initialization is $\mathcal{O}(KNl^2 + d^2)$; the complexity of the main loop is $T \times \mathcal{O}((N_S + N_D)K(d^2 + ld + ld^2) + d^3)$. The overall complexity for HM3L algorithm is $\mathcal{O}(T(N_S + N_D)Kld^2)$. The complexity analysis for KHM3L algorithm can be similarly done and the overall complexity is $\mathcal{O}(T(N_S + N_D)KNd^2)$.

5.8 Conclusions

In this chapter, we proposed linear and kernelized hierarchical multimodal metric learning algorithm which can efficiently learn multiple metrics for multi-modal data while fully exploring the relationships among these metrics. Experimental results on four datasets show that the proposed metric learning algorithm outperforms other metric learning algorithms dealing with multi-modal data and provide the best performance for all the experiments considered. We view feature learning as a different problem and only focus on learning discriminative metrics for multimodal data in order to improve the multimodal classification accuracy. As we separate the feature learning process from the metric learning process, the proposed approach is quite general and can be applied to many different applications with many different feature types. Especially, since many computer vision and image processing problems involve dealing with multiple descriptors and thus can be considered in multi-modal settings, the proposed algorithms can be applied where appropriate distance metrics are required and can boost the performance of related tasks.

Chapter 6: Conclusions and Future Research

This dissertation was initially motivated by the challenges in building active authentication system on mobile platforms and was further extended to exploring general multimodal recognition (classification) problems which arise in many computer vision and machine learning problems.

The Active Authentication dataset (UMDAA) we built became a useful resource for studying touch data and face images for active authentication problems. In Chapter 2, we designed kernel sparse representation-based classifier and kernel dictionary learning-based classifers for touch gestures. Experiments on screen touch data of UMDAA datasets as well as two publicly available screen touch datasets showed that the kernel dictionary can be a potential signature for user authentication on mobile platforms. Cross-session experiments showed a significant drop in the performance of all the methods. This problem can be viewed as domain adaptation problem which was addressed in Chapter 3.

In Chapter 3, we proposed a sparsity-based framework for solving the domain adaptation problems. The proposed DASRC algorithm is applicable to single-source domain, multi-source and heterogeneous domain adaptation problems. We proposed an iterative algorithm consisting of the ADMM method and the SOC method for solving the optimization problem. Extensive experiments on the UMDAA dataset showed that our method can perform better than many state-of-the-art domain adaptation methods.

After considering screen touch data and face data separately, we focused on developing efficient fusion algorithms in order to provide better performance using multiple sources of data than using any single source of data by itself. In Chapter 4, we proposed multi-task, multivariate low-rank and joint sparse representation-based methods for multimodal recognition. Our methods can be viewed as a generalized version of multivariate low-rank and joint sparse regression, where low-rank and joint sparse constraints are imposed across all the modalities. We further explored common representation across all modalities in order to get a more robust representation at some cost of losing information. Efficient algorithms based on ADMM were derived to solve the proposed problems and extensive experimental results on UMDAA dataset as well as other multimodal recognition datasets demonstrated the robustness and effectiveness of the proposed algorithms.

In Chapter 5, we proposed noval multimodal metric learning algorithm and its kernel extension which can learn multiple metrics simultaneously for multimodal data in order to improve multimodal classification performance. The proposed formulation takes into account both the different characteristics exhibited in different modalities and the relationship among the multiple metrics. Experimental results for tagged image classification, GBD object recognition and the RGBD scene recognition problems showed that the proposed metric learning algorithm outperformed other metric learning algorithms dealing with multi-modal data and provide the best performance for all the experiments considered.

The multimodal learning algorithms proposed in Chapter 4 and Chapter 5 are general and are suitable for many different applications with many different feature types. Since many computer vision and machine learning problems involve dealing with multiple descriptors and thus can be considered in multi-modal settings, the proposed multimodal

learning algorithms can boost the performance of related tasks.

In the future, we plan to study the interactions and inferences among different modalities in multimodal problems which arise in many applications and are receiving a lot of attention [133] [134] [135]. A few specific problems of interest are as follows. (1) We would like propose efficient algorithms to transfer knowledge learned from one modality to other modalities to improve classification, clustering and retrieval performance. (2) Similar to image captioning [135], we would like to explore models to generate data of one modality from other modalities.

Bibliography

[1] R. P. Guidorizzi. Security: Active authentication. *IEEE IT Professional Magazine*, 15(4):4–7, 2013.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, December 2003.

[3] Eric P. Xing, Michael I. Jordan, Stuart J Russell, and Andrew Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*.

[4] A. Serwadda, V.V. Phoha, and Z. Wang. Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms. In *IEEE International Conference on Biometrics: Theory, Applications and Systems*, pages 1–8, Sept 2013.

[5] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song. Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication. *IEEE Transactions on Information Forensics and Security*, 8(1):136–148, Jan 2013.

[6] Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 2879–2886, 2012.

[7] V. Štruc and N. Pavešić. The complete gabor-fisher classifier for robust face recognition. *EURASIP Advances in Signal Processing*, 2010:26, 2010.

[8] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *Proceedings of the 3rd International Conference on Analysis and Modeling of Faces and Gestures*, AMFG'07, pages 168–182, 2007.

[9] John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, February 2009.

[10] M. E. Fathy, V. M. Patel, and R. Chellappa. Face-based active authentication on mobile devices. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1687–1691, April 2015.

[11] H. Zhang, V. M. Patel, M. Fathy, and R. Chellappa. Touch gesture-based active user authentication using dictionaries. In *IEEE Winter Conference on Applications of Computer Vision*, pages 207–214, Jan 2015.

[12] S. S. S. Crihalmeanu, A. Ross, and L. Hornak. A protocol for multibiometric data acquisition, storage and dissemination. Technical report, WVU, Lane Department of Computer Science and Electrical Engineering, 2007.

[13] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, pages 15–29, Berlin, Heidelberg, 2010. Springer-Verlag.

[14] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824, May 2011.

[15] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, June 2015.

[16] Lingjun Li, Xinxin Zhao, and Guoliang Xue. Unobservable re-authentication for smartphones. In *Network & Distributed System Security Symposium*, Feb 2013.

[17] Tao Feng, Ziyi Liu, Kyeong-An Kwon, Weidong Shi, B. Carbunar, Yifei Jiang, and N. Nguyen. Continuous mobile authentication using touchscreen gestures. In *IEEE Conference on Technologies for Homeland Security*, pages 451–456, Nov 2012.

[18] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009.

[19] J. K. Pillai, V. M. Patel, R. Chellappa, and N. K. Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1877–1893, Sept 2011.

[20] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition. In *ECCV (4)'10*, pages 1–14, 2010.

[21] H. Van Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Design of non-linear kernel dictionaries for object recognition. *IEEE Transactions on Image Processing*, 22(12):5123–5135, 2013.

[22] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers*, 1993.

[23] B. Scholkopf and A. J. Smola. *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.

[24] M. Aharon, M. Elad, and A. M. Bruckstein. The k-svd: an algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Transaction on Signal Process.*, 54(11):4311–4322, 2006.

[25] V. M. Patel, W. Tao, S. Biswas, P. J. Phillips, and R. Chellappa. Dictionary-based face recognition under variable lighting and pose. *IEEE Transactions on Information Forensics and Security*, 7(3):954–965, 2012.

[26] Jing Jiang. *Domain adaptation in natural language processing*. PhD thesis, University of Illinois at Urbana-Champaign, 2008.

[27] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[28] Hal Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[29] Wen Li, Lixin Duan, Dong Xu, and IW. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, June 2014.

[30] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2288–2302, 2014.

[31] Boqing Gong, Yuan Shi, Fei Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.

[32] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, volume 6314, pages 213–226, 2010.

[33] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1785–1792, 2011.

[34] I-Hong Jhuo, Dong Liu, D.T. Lee, and Shih-Fu Chang. Robust visual domain adaptation with low-rank reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2175, 2012.

[35] Jun Yang, Rong Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *ACM Multimedia*, pages 188–197. ACM, 2007.

[36] Lixin Duan, Ivor Wai-Hung Tsang, Dong Xu, and Stephen J. Maybank. Domain transfer svm for video concept detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1375–1381, 2009.

[37] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. In *International Conference on Learning Representations*, 2013.

[38] Lixin Duan, I.W. Tsang, and Dong Xu. Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479, 2012.

[39] S. Wang, L. Zhang, Liang Y., and Q. Pan. Semi-coupled dictionary learning with applications in image super-resolution and photo-sketch synthesis. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[40] Sumit Shekhar, Vishal M. Patel, Hien V. Nguyen, and Rama Chellappa. Generalized domain-adaptive dictionaries. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[41] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *European Conference on Computer Vision*, volume 7575, pages 631–645, 2012.

[42] Jie Ni, Qiang Qiu, and Rama Chellappa. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In *IEEE International Conference on Computer Vision*, 2013.

[43] Jing Jiang. A literature survey on domain adaptation of statistical classifiers. *Tech report*, 2008.

[44] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[45] Vishal M. Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 2014.

[46] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, Nov 2013.

[47] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, Jan 2011.

[48] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, Feb 2014.

[49] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

[50] Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation, 2012.

[51] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[52] P. K. Varshney. Multisensor data fusion. *Electronics and Communication Engineering Journal*, 9(6):245–253, 1997.

[53] A. Ross and A. K. Jain. Multimodal biometrics: an overview. In *European Signal Processing Conference*, pages 1221–1224, 2004.

[54] A. Ross, K. Nandakumar, and A. K. Jain. *Handbook of Multibiometrics*. Springer, 2006.

[55] Stephen Becker, Jérôme Bobin, and Emmanuel J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. Img. Sci.*, 4:1–39, Jan 2011.

[56] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[57] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in Neural Information Processing Systems 22*, pages 2080–2088. 2009.

[58] M. Golbabaee and P. Vandergheynst. Hyperspectral image compressed sensing via low-rank and joint-sparse matrix recovery. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2741–2744, March 2012.

[59] Pierre-Andr Savalle, Emile Richard, and Nicolas Vayatis. Estimation of simultaneously sparse and low rank matrices. In *International Conference on Machine learning*, 2012.

[60] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[61] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 2013.

[62] Yu-Xiang Wang, Huan Xu, and Chenlei Leng. Provable subspace clustering: When lrr meets ssc. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 64–72. 2013.

[63] S. Shekhar, V. M. Patel, N. M. Nasrabadi, and R. Chellappa. Joint sparse representation for robust multimodal biometrics recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):113–126, Jan 2014.

[64] H. Zhang, V. M. Patel, and R. Chellappa. Robust multimodal recognition via multitask multivariate low-rank representations. In *IEEE International Conference on Automatic Face and Gesture Recognition*, volume 1, pages 1–8, May 2015.

[65] A. Rattani, D. Kisku, M. Bicego, and M. Tistarelli. Feature level fusion of face and fingerprint biometrics. In *IEEE International Conference on Biometrics: Theory, Applications, and Systems*, pages 1–6, 2007.

[66] X. Zhou and B. Bhanu. Feature fusion of face and gait for human recognition at a distance in video. In *International Conference on Pattern Recognition*, pages 529–532, 2006.

[67] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, July 2011.

[68] X.-T. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3493–3500, 2010.

[69] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang. Multiobservation visual recognition via joint dynamic sparse representation. In *International Conference on Computer Vision*, pages 595–602, 2011.

[70] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–67, Feb 2006.

[71] H. Zhang, V. M. Patel, and R. Chellappa. Multitask multivariate common sparse representations for robust multimodal biometrics recognition. In *IEEE International Conference on Image Processing*, pages 202–206, Sept 2015.

[72] M. Dao, N. H. Nguyen, N. M. Nasrabadi, and T. D. Tran. Collaborative multi-sensor classification via sparsity-based representation. *IEEE Transactions on Signal Processing*, 64(9):2400–2415, May 2016.

[73] Soheil Bahrampour, Nasser M. Nasrabadi, Asok Ray, and William Kenneth Jenkins. Multimodal task-driven dictionary learning for image classification. *IEEE Trans. Image Processing*, 25(1):24–38, 2016.

[74] Ashish Shrivastava, Mohammad Rastegari, Sumit Shekhar, Rama Chellappa, and Larry S. Davis. Class consistent multi-modal fusion with binary features. June 2015.

[75] J. S. Turek, J. Sulam, M. Elad, and I. Yavneh. Fusion of ultrasound harmonic imaging with clutter removal using sparse signal separation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 793–797, April 2015.

[76] Mohammad Rastegari, Ali Farhadi, and David Forsyth. Attribute discovery via predictable discriminative binary codes. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, 2012.

[77] J. Wright, AY. Yang, A Ganesh, S.S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, Feb 2009.

[78] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, November 2010.

[79] M. Elad. *Sparse and Redundant Representations: From theory to applications in Signal and Image processing*. Springer, 2010.

[80] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956–1982, Mar 2010.

[81] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), May 2011.

[82] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[83] Sharat Chikkerur, Chaohang Wu, and Venu Govindaraju. A systematic approach for feature extraction in fingerprint images. In *Biometric Authentication, First International Conference, ICBA 2004, Hong Kong, China, July 15-17, 2004, Proceedings*, pages 344–350, 2004.

[84] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti. Filterbank-based fingerprint matching. *Trans. Img. Proc.*, 9(5):846–859, May 2000.

[85] S.J. Pundlik, D.L. Woodard, and S.T. Birchfield. Non-ideal iris segmentation using graph cuts. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1–6, June 2008.

[86] Peter Kovesi Libor Masek. Matlab source code for a biometric identification system based on iris patterns, 2003.

[87] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, 2010.

[88] B. Browatzki, J. Fischer, B. Graf, H. H. Blthoff, and C. Wallraven. Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1189–1195, Nov 2011.

[89] A. Janoch, S. Karayev, Yangqing Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1168–1174, Nov 2011.

[90] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, pages 746–760, 2012.

[91] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision*, pages 1625–1632, Dec 2013.

[92] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 902–909, June 2010.

[93] Ruofei Zhang, Zhongfei Zhang, Mingjing Li, Wei-Ying Ma, and Hong-Jiang Zhang. A probabilistic semantic model for image annotation and multimodal image retrieval. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 846–851 Vol. 1, Oct 2005.

[94] Pengcheng Wu, Steven C.H. Hoi, Hao Xia, Peilin Zhao, Dayong Wang, and Chunyan Miao. Online multimodal deep similarity learning with application to image retrieval. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 153–162, New York, NY, USA, 2013. ACM.

[95] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 689–696, 2011.

[96] Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.

[97] K.Q. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.

[98] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, Corvalis, Oregon, USA, 2007.

[99] Ratthachat Chatpatanasiri, Teesid Korsrilabutr, Pasakorn Tangchanachaianan, and Boonserm Kijsirikul. A new kernelization framework for mahalanobis distance learning algorithms. *Neurocomput.*, 73(10-12):1570–1579, jun 2010.

[100] Jun Wang, Huyen T. Do, Adam Woznica, and Alexandros Kalousis. Metric learning with multiple kernels. In *Advances in Neural Information Processing Systems 24*, pages 1170–1178. 2011.

[101] Shai Shalev-Shwartz, Yoram Singer, and Andrew Y. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, 2004.

[102] Prateek Jain, Brian Kulis, and Inderjit S. Dhillon. Inductive regularized learning of kernel functions. In *Advances in Neural Information Processing Systems 23*, pages 946–954. 2010.

[103] A. Wang, J. Cai, J. Lu, and T. J. Cham. Mmss: Multi-modal sharable and specific feature learning for rgb-d object recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1125–1133, Dec 2015.

[104] Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu. Multi-task low-rank metric learning based on common subspace. In *Neural Information Processing - 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part II*, pages 151–159, 2011.

[105] B. McFee and G. R. G. Lanckriet. Metric learning to rank. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*, June 2010.

[106] A. Frome, Y. Singer, Fei Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.

[107] Karen Simonyan, Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. In *British Machine Vision Conference, BMVC 2013, Bristol, UK, September 9-13, 2013*, 2013.

[108] Sumit Chopra, Raia Hadsell, and Yann Lecun. Learning a similarity metric discriminatively, with application to face verification. In *In Proc. of Computer Vision and Pattern Recognition Conference*, pages 539–546. IEEE Press, 2005.

[109] Jason V. Davis and Inderjit S. Dhillon. Structured metric learning for high dimensional problems. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 195–203, New York, NY, USA, 2008. ACM.

[110] Wei Liu, Cun Mu, Rongrong Ji, Shiqian Ma, John R. Smith, and Shih-Fu Chang. Low-rank similarity metric learning in high dimensions. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2792–2799, 2015.

[111] Yiming Ying, Kaizhu Huang, and Colin Campbell. Sparse metric learning via smooth optimization. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 2214–2222. Curran Associates, Inc., 2009.

[112] Rómer Rosales and Glenn Fung. Learning sparse metrics via linear programming. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 367–373, 2006.

[113] Wei Liu, Shiqian Ma, Dacheng Tao, Jianzhuang Liu, and Peng Liu. Semi-supervised sparse metric learning using alternating linearization optimization. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 1139–1148, 2010.

[114] Daryl KH Lim, Brian McFee, and Gert Lanckriet. Robust structural metric learning. In *International Conference on Machine Learning*, 2013.

[115] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013.

[116] Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2012.

[117] Pengtao Xie and Eric P Xing. Multi-modal distance metric learning. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1806–1812. AAAI Press, 2013.

[118] H. Zhang, T. Huang, N. Nasrabadi, and Y. Zhang. Heterogeneous multi-metric learning for multi-sensor fusion. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8, July 2011.

[119] Junlin Hu, Jiwen Lu, Junsong Yuan, and Yap-Peng Tan. Large margin multi-metric learning for face and kinship verification in the wild. In *Computer Vision 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014*, pages 252–267, 2014.

[120] Zhen Cui, Wen Li, Dong Xu, Shiguang Shan, and Xilin Chen. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 3554–3561, 2013.

[121] Stephen Boyd and Almir Mutapcic. Stochastic subgradient methods, 2007.

[122] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. NUS-WIDE: a real-world web image database from national university of singapore. In *Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR 2009, Santorini Island, Greece, July 8-10, 2009*, 2009.

[123] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Kernel descriptors for visual recognition. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 244–252, 2010.

[124] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 821–826, Sept 2011.

[125] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[126] M. Blum, Jost Tobias Springenberg, J. Wlfing, and M. Riedmiller. A learned feature descriptor for object recognition in rgb-d data. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1298–1303, May 2012.

[127] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Unsupervised feature learning for RGB-D based object recognition. In *Experimental Robotics - The 13th International Symposium on Experimental Robotics, ISER 2012, June 18-21, 2012, Québec City, Canada*, pages 387–402, 2012.

[128] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems 27*, pages 487–495. 2014.

[129] Anran Wang, Jianfei Cai, Jiwen Lu, and Tat-Jen Cham. Modality and component aware feature fusion for rgb-d scene classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[130] Hongyuan Zhu, Jean-Baptiste Weibel, and Shijian Lu. Discriminative multimodal feature fusion for rgbd indoor scene recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[131] Yiyi Liao, Sarath Kodagoda, Yue Wang, Lei Shi, and Yong Liu. Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 2318–2325, 2016.

[132] B. McFee and G.R.G. Lanckriet. Learning multi-modal similarity. *Journal of Machine Learning Research*, 12:491–523, February 2011.

[133] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, MarcAurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *Neural Information Processing Systems (NIPS)*, 2013.

[134] Ran Xu, Caiming Xiong, Wei Chen, and Jason J. Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2346–2352, 2015.

[135] Andrej Karpathy and Fei-Fei Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137, 2015.