ON REPRESENTATION BASED PATTERN CLASSIFICATION MODELS

ZHU PENGFEI

Ph.D

The Hong Kong

Polytechnic University

2015

The Hong Kong Polytechnic University
Department of Computing

On Representation Based Pattern Classification Models

Pengfei Zhu

A thesis submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

September 2014

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

Pengfei Zhu (Name of student)

Abstract

In computer vision and pattern recognition, there are a variety of image based classification tasks, e.g., face recognition, action recognition, object recognition, texture classification, handwritten digit recognition, etc. How to choose a suitable classifier for the given classification task is not a trivial problem, and it depends on data type, data distribution, data size, and feature property. According to "no free lunch" theorem in machine learning, there is no one classifier that can always achieve the state-of-the-art performance in all classification tasks. Intuitively, a robust, efficient, and scalable classifier with good understandability, scalability and generalization ability is always desired.

Representation based classification has been widely used in pattern classification and achieves superior performance. It is based on the assumption that a query sample can be more accurately approximated by a linear combination of training samples of its class than other classes. Many representation based classification models have been developed, including sparse/collaborative representation, low-rank representation, robust representation, kernel representation, generic representation, multi-modal/cross-modal representation, etc. Representation residuals in these models are discriminative and a query sample can be classified to the class with the minimal reconstruction residual. Meanwhile, representation coefficients can also be used as features to enhance classification. In addition, in middle-level feature extraction, in contrast to vector quantization, sparse coding can be introduced to obtain a soft representation for classification.

Although representation based classification models have achieved a great success in different classification tasks, there are still many problems remaining. When

there are only a small number of training samples, the representation tends to be over-determined and therefore the query sample may not be well represented. When the number of the training samples is very large, the time complexity and memory consumption of representation based classifiers becomes a challenging issue. Besides, the existing representation based classifiers are mostly designed to accomplish single image based classification tasks. However, for video based face recognition and multi-view object recognition, the task becomes an image set classification problem. It is demanded to extend representation based classifiers from image based to image set based models. Finally, most existing representation based classifiers are non-discriminative in the representation process. It is interesting to investigate if the samples can be projected to a discriminative feature space to enhance the classification performance.

In this thesis, we aim to develop new representation based classification models for small sample size problems, big sample size problems, image set classification problems, and discriminative representation problems, respectively.

In Chapter 2, to solve the small sample size problem in face recognition, a patch based collaborative representation classifier (PCRC) is proposed. Both the query and gallery face images are divided into patches and then the query patch is represented by the gallery patch dictionary. Classification outputs of all the patches are combined by majority voting to get the final output. As PCRC is sensitive to patch size, a multi-scale PCRC is proposed to fuse the classification outputs of different path sizes by margin distribution optimization.

In Chapter 3, a local generic representation (LGR) based approach is proposed for face recognition with single sample per person. A generic intra-class variation

dictionary is constructed from a generic dataset, and it can well compensate for the face variations lacked in the gallery set. A correntropy based metric is adopted to measure the loss of each patch so that the importance of different patches in face recognition can be more robustly evaluated.

In Chapter 4, a self-representation induced classifier (SRIC) is proposed for representation with big sample size. Different from the existing sample-level representation, we proposed representation based classifiers from the perspective of feature-level representation. The time complexity of SRIC is only related with feature dimension and the number of classes. Hence, it is very suitable for classification tasks with a large amount of training samples and a small number of classes.

In Chapter 5, an image set based collaborative representation model is proposed for image set based face recognition. Considering the distinctiveness of samples in the query image set and the correlation between the gallery image sets, we model both the query and gallery image set as hulls. Then the hull of the query image set is collaboratively represented on the gallery image sets. Regularized hull and kernel convex hull are both considered to develop robust image set based collaborative representation classifiers.

In Chapter 6, by considering representation based classifiers as point-to-set distance based classifiers, we extended distance metric learning from point-to-point distance to point-to-set and set-to-set distance. The metric learning problem is modeled as a sample pair classification task and can be efficiently solved by standard support vector machine solvers.

To sum up, in this thesis we developed patch based collaborative representation, local generic representation, regularized self-representation, image set based col-

laborative representation, and point-to-set/set-to-set distance metric learning methods to address the representation problems with small sample size, big sample size, and image sets for pattern recognition, respectively. Our extensive experimental results demonstrated the state-of-the-art performance of the proposed methods. In the future work, we will investigate generic dictionary learning for face recognition in the wild, cross-modal/multi-modal dictionary learning and metric learning methods under the representation based pattern classification framework.

Publications

The following papers, published, in press or submitted, are the partial outputs of my PhD studies in PolyU.

1. **P. Zhu**, W. Zuo, L. Zhang, Q. Hu, S. Shiu, "Unsupervised Feature Selection by Regularized Self-representation," Pattern Recognition. vol.48, no.2, pp. 438-446, Feb. 2015

2. **P. Zhu**, M. Yang, L. Zhang, "Local Generic Representation for Face Recognition with Single Sample per Person," In Proc. ACCV 2014.

3. **P. Zhu**, W. Zuo, L. Zhang, S. Shiu, D. Zhang, "Image Set based Collaborative Representation for Face Recognition," IEEE Tran. on Information Forensics and Security. vol. 9 no. 7 pp. 1120-1132 Jul. 2014.

4. **P. Zhu**, L. Zhang, W. Zuo, D. Zhang, "From Point to Set: Extend the Learning of Distance Metrics," In Proc. ICCV 2013.

5. **P. Zhu**, L. Zhang, Q. Hu, S. Shiu, "Multi-scale Patch based Collaborative Representation for Face Recognition with Margin Distribution Optimization," In Proc. ECCV 2012.

6. L. Zhang, **P. Zhu**, Q. Hu, D. Zhang, "A Linear Subspace Learning Approach via Sparse Coding, " In Proc. ICCV 2011.

7. **P. Zhu**, L. Zhang, W. Zuo, X. Feng, "A Discriminative Self-representation induced Classifier." Under Preparation.

Acknowledgements

I am deeply grateful toward to my chief supervisor Prof. Lei Zhang. His constant and gentle guidance has allowed me to enjoy the research I have done. He is always there whenever I need help, and with his support and encouragement, I overcame the difficulties in this work. I am thankful for all his insightful comments and for the very thorough revision he has made of my articles and this thesis. I consider myself very fortunate to have the chance to learn from him.

I would like to thank my co-supervisor Dr. Simon C. K. Shi for his valuable and helpful suggestions to my research.

I was fortunate to get to know Prof. Wangmeng Zuo, Prof. Xiangchu Feng, Prof. Peihua Li, Prof. Liang Lin and Prof. Shiguang Shan in my Ph.D. study. Some ideas in my research work directly originated from the discussion with them, and their excellent academic experiences greatly benefit me in doing good research. I have greatly enjoyed working with my colleagues: Lin Zhang, Peng Bo, Jin Xie, Kaihua Zhang, Meng Yang, Zhizhao Feng, Shuhang Gu, Xingzheng Wang, Feng Liu, Xiaofeng Qu, Kunai Zhang, Faqiang Wang, Zhaoxin Li. I am thankful to their share of time with help and support over the period of my Ph.D. program.

There are many other friends I've met along the way with whom I've shared part of this road. To all of you I'm thankful for sharing the thoughts and perspectives.

At last, I would express my deep gratitude to my parents, for their endless love and care through out my life. Thanks for the most special support and love from my wife, Tianjiao Zhao. No matter how far away, they are always my biggest and best supporters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Classification tasks

In our daily life, human beings need to get the identity of one person, search related text, audio, pictures or videos, distinguish salmon from bass, etc. Fortunately, all these needs can be satisfied via classification. In computer vision and pattern recognition, there are various classification tasks. As shown in Fig. 1.1, the classification tasks include face/iris/palmprint/fingerprint/finger-knuckle recognition, action recognition, texture classification, image classification, handwritten digit recognition, etc. For a general classification task, there are four crucial steps: data collection, data preprocessing, feature extraction and classification. Among all the four steps, feature extraction and classification has attracted much attention of researchers in the past few years.

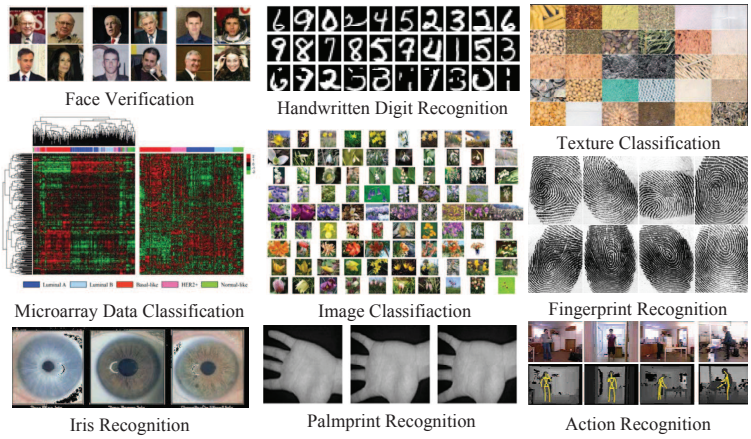


Figure 1.1 Classification tasks in computer vision and pattern recognition.

1.1.1 Feature extraction

The main purpose of feature extraction is to produce good representations for data, which can be used for detection, recognition, prediction, or visualization. Successful feature extraction algorithms should eliminate irrelevant variabilities of the input data, while preserving the useful information for the ultimate task. Feature extraction methods can be categorized into three types: subspace learning, local features and feature learning.

Subspace learning As a popular dimensionality reduction and feature extraction technique, subspace learning has been successfully used in various computer vision and pattern recognition applications, for example, appearance based face recognition (FR). Representative subspace learning methods include principal component analysis (PCA), e.g., Eigenface [180], Fisher linear discriminant analysis (FLDA) [11], the manifold learning [176] [156] based locality preserving projection (LPP) [75], local discriminant embedding (LDE) [25], graph embedding [207], etc. According to if the class label information of the training samples is exploited, the

linear subspace learning methods can be categorized into unsupervised methods (e.g., PCA and LPP) and supervised methods (e.g., FLDA [11], regularized LDA (RLDA) [114] and LDE). Subspace learning methods are not limited by data types.

Local features In computer vision and multimedia tasks, images are the most common data type. Intuitively, intensity features can be directly used for different tasks. However, the poor performance of intensity features and the undiscovered hidden information in the image drive researchers to extract more useful features from images. Local features are distinctive and invariant to many kinds of geometric and photometric transformations [108]. For a local feature, it consists of a feature detector and a feature descriptor. Feature detectors need to detect the key points and regions of an image. The history of feature detector can be tracked back to the Moravec's corner detector [130], and from then on a large number of corner and region detectors [70][126][181][92] have been proposed. After key points and regions are detected, local descriptors are used for feature description. We can categorize existing descriptors into filter-based descriptors (e.g., steerable filters [49], Gabor filters [121] and complex filters [159]), distribution-based descriptors, (e.g., SIFT, LBP, shape context, and GLOH), textons [107] and derivative-based descriptors [48]. Different descriptors may be optimal for different tasks. Hence, it is quite necessary to design a proper local feature for different tasks.

Feature learning The model of visual cortex suggests that the brain of human beings extracts features from edge, patch, surfaces, and then to objects [84, 149, 164]. The observation and decision process is usually a hierarchy of representations with increasing level of abstraction and each level is a trainable feature transform. Besides image classification, there is a pipeline for text classification, i.e., from Char-

acter, word, word group, clause, sentence to story [100]. In speech recognition, the similar process goes from sample, spectral band, sound, phone, phoneme to word [76]. In machine learning and artificial intelligence, how can we learn feature hierarchies? In neuroscience, the way how cortex learns perception needs to be explored. Fortunately, deep Learning develops a hierarchy of deep architecture to address these problem. Deep learning constructs a hierarchy of trainable transforms, from low-level features that shared among categories to more global and more invariant high-level features. There are three deep architectures, i.e., feed-forward (e.g., multi-layer neural nets [2], convolutional nets [100]), feed-back (stacked sparse coding [221], deconvolutional nets [224]) and bi-directional (e.g., deep boltzmann machines [158], stacked auto-encoders [58]). Additionally, there are three types of training protocols, including fully supervised, unsupervised layerwise training plus supervised classifier on top, unsupervised layerwise training plus global supervised fine-tuning. To learn invariant feature, the overall architecture is composed of normalization, filter bank, non-linearity and pooling. There are two types of normalization, subtractive (e.g., average removal, high pass filtering) and divisive (e.g., local contrast normalization, variance normalization). The non-linearity can be introduced by non-linear dimension expansion or sparse non-linear expansion. Finally, by pooling, semantically similar regions can be brought together. As deep learning can extract invariant features and is consistent with the cortex of human brains, it has been successfully used in pedestrian detection [163], image segmentation [37], action recognition [88], scene parsing [170], speech recognition [76], etc.

1.1.2 Classifiers

After features are extracted from text, image, audio or video, suitable classifiers should be chosen for classification. According to the number of labels, there are single-label and multi-label classification tasks. Besides, according to the number of modalities, there are single modal classifiers, multi-modal classifiers and cross modal classifiers. According to the availability of the training labels, the classifications tasks can be also categorized into weak-supervised, semi-supervised and supervised tasks. In this following part, the popular classifiers are categorized and reviewed.

Distance/similarity based classifiers Given two samples, we need to measure their similarity/dissimialrity to judge whether they belong to the same object [82]. Given a query sample, the distance from the query sample to the training samples is also needed to get the identification. For both identification and verification problem, a proper distance metric should be designed or learned for a certain task. K nearest neighbor classifier (KNN) is one of the most popular and efficient classifiers in pattern recognition. KNN assigns the query sample to the class with the largest frequency in the k-nearest neighbors. There are two factors that affect the performance of KNN, i.e., distance metric and K. The distance metrics can be Euclidian distance, cosine distance, Manhattan Distance, Mahalanobis distance, etc. In recent years, it has been increasingly popular to learn a desired distance metric from the given training samples in many visual classification tasks, such as face/action/kinship verification [66], visual tracking [89], and image retrieval [1]. Metric learning methods can be categorized into unsupervised [33], semi-supervised [1] and supervised ones [1, 66, 89], according to the availability of the class labels of training samples. As

the naive linear search over all the training data vectors are quite time-consuming, two branch and bound search algorithms, i.e., kd-trees [50] and ball trees [52] are introduced to accelerate the searching process. Then to alleviate the impact of curse of dimensionality, hashing technique was proposed to build search structure for performing similarity search over high-dimensional data [61].

Rule based classifiers In some applications, such as medical analysis, stock prediction and fault diagnosis, the understandability of classifiers is quite important. The users need to get definite rules for analysis or diagnosis. The most popular rule based classifiers are decision trees, e.g., classification and regression tress [152] and C4.5 [146]. Attentions are also paid to extract rules from block box classifiers, such as support vector machines [137] and artificial neural networks [177]. Rule based classifiers are composed of two parts: rule extraction, and rule pruning. More detailed discussions about rule learning can be found in [53].

Linear/nonlinear discriminant classifiers Given a query sample $\boldsymbol{x}$, it can be classified by a discriminant function $f(\boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x} + b$. For non-linear cases, with kernel mapping, the discriminant function becomes $f(\boldsymbol{x}) = \boldsymbol{w}\phi(\boldsymbol{x}) + b$. This kind of classifier assume the samples of different classes can be separated by a series of classification hyperplane. Support vector machines [184], linear regression [161], and logistic regression [77] can be categorized into this kind of classifier. Ensemble learning methods, e.g., boosting, also belong to linear discriminant classifiers. Additionally, to deal with multi-task problems, multi-kernel learning extends nonlinear discriminant classification model to the multi-task case [7].

Representation based classifiers Inspired from the fact natural images can be generally coded by structural primitives and these primitives are qualitatively similar to

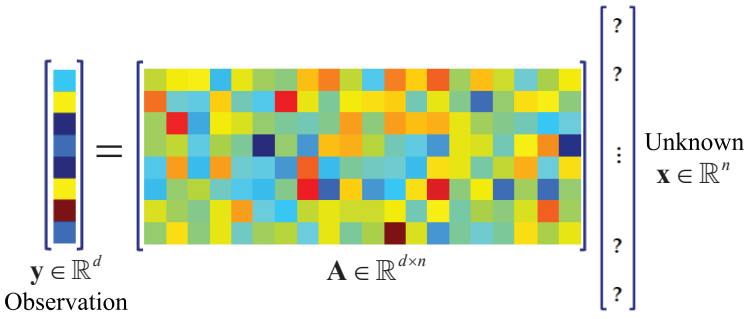


Figure 1.2 An under-determined linear system.

simple cell receptive fields, Olshausen et al. proposed to represent a natural image using a small number of basis functions from an over-complete code set [138, 139]. As shown in Fig. 1.2, given a signal $\boldsymbol{y}$ and a set of bases $\boldsymbol{A}$, $\boldsymbol{y}$ can be represented as a linear combination of bases, i.e., $\boldsymbol{Ax}$. Finding a good representation has been the topic of many applications, e.g., signal reconstruction [28], image restoration [116], etc. Besides applications in low-level vision, representation based models have also been used in high-level image classification tasks [201, 212]. Firstly, the representation residuals can be used for classification [201, 226]. The representation residual of each class has discrimination ability and hence can be used for classification. Secondly, the representation coefficients vector are used as the feature, and then the traditional classifiers (e.g., SVM) are utilized for the final classification outputs [212]. As there are noises, non-linear data structure and multi-modalities, robust [214], kernel [55], cross-modal [194] and multi-modal representations [219, 223] are proposed for classification for each case, respectively. To pursue a good representation, the bases, also called dictionary, are quite important in representation learning. In Section 1.2, we will review some representation based models.

1.2 Overview of representation based pattern classification

In this section, we review the representation learning models. Firstly, the representation based classifiers are reviewed. Besides, dictionary learning methods are also reviewed.

1.2.1 Representation based classifiers

As there are different modalities of objects in nature, there are single modal, multi-modal and cross-modal representation based classification models.

Single-modal representation

For single-modal representation, a series of models have been proposed, from sparse representation, collaborative representation to robust representation, kernel representation and generic representation.

Sparse representation Nowadays, the data are increasingly massive and high-dimensional. How can we find the low-dimensional structure from such high-dimensional data? Because of rich local regularities, global symmetries, repetitive patterns, or redundant sampling, visual data usually exhibits low-dimensional structures. In section 1.1.2, the mechanism of representation based classification models has been introduced. To find a good representation of a given signal, efficient prior should be imposed. Based on the observation of the representation and cognitive science, in many cases, the presentation coefficients are sparse. Sparse representation has been widely used image restoration [117], image resolu-

tion [211], visual tracking [124], image classification [212], etc. For example, a query face can be sparsely reconstructed by only several related faces. In [201], a sparse representation based classifier (SRC) is proposed for face recognition. It uses the reconstruction residual of each class for classification. Actually, SRC can be considered as an extension from nearest neighbor classifier, nearest feature line, nearest subspace classifier. The difference is that the representation is done on the training samples of all classes. Intuitively, l_0-norm is used to measure the sparsity. However, it is non-convex and a NP hard optimization problem. l_1-norm is the most strictly convex hull of l_0-norm and it is convex,though it is not smooth as l_2-norm. Hence, l_1-norm is introduced and it is a convex optimization problem. There are quite a lot of l_1-norm optimization algorithms, i.e., primal-dual interior-point, homotopy, gradient projection, iterative thresholding, proximal gradient and augmented lagrangian Methods [209]. Usually, sparse presentation emphasizes the sparsity of representation coefficients. Sometimes, in case of corruptions and occlusions in the face image, the representation residual is also measured by l_1-norm to be robust to noise. The fidelity measure is up to the representation error distribution. If it satisfies the Gaussian distribution, then l_2-norm is adopted. Otherwise, if it satisfies the Laplacian distribution, e.g., face images with pix corruption, l_1-norm should be used.

Collaborative representation In representation based classifiers, suitable regularization should be imposed on the representation coefficients according to the prior knowledge about the solution. If we know in advance that the solution is sparse, i.e., only a few elements are relevant, then l_1-norm can be well adopted, which leads to a lasso problem. However, should all the representation models be regularized

by sparsity norms? What does the succuss of SRC own to? In [226], a collaborative representation based classifier (CRC) is proposed by replacing the l_1-norm in SRC with l_2-norm. Without solving a time-consuming l_1-norm optimization problem, CRC only needs to solve a ridge regression problem that has a closed-form solution. Whereas, CRC achieves comparable recognition performance to SRC while with much lower computation consumption. Besides, CRC explains the success of SRC from the perspective of collaborative representation, which means that the across subject face similarity can be used to help represent faces of other persons. CRC can explains the superior performance of SRC in face recognition. However, the concept of collaborative representation does not necessarily apply to all the classification tasks. The debate of l_1 and l_2-norm regularization induces more discussions and experimental validations. Overall, if samples are well conditioned, the dense representation can lead to comparable performance. In the other case, when samples are highly coherent, sparse representation is more discriminative. Besides l_1-norm and l_2-norm regularization, there maybe exists the structural relationship in the data. Hence, other regularization, e.g., group lasso, can be introduced to regularize the representation coefficients. If we can discover the latent structure and prior knowledge from the data itself, we can choose proper regularization.

Robust representation In face recognition, there are usually corruptions, occlusions or disguises in face images. Then robust face recognition algorithms have been proposed to deal with various noise. In the raw SRC, when there are pixel corruptions, l_1-norm is used to measure the representation error. The key motivation of robust representation is to alleviate the impact of the corrupted pixels. The solution is to find a proper measure for the representation residual. What is the

distribution of representation error, Gaussian or Laplacian? In real applications, the distributions are diverse. In [214], motivated by maximum a posterior (MAP), a robust sparse coding (RSC) algorithm is proposed by iterative reweighted strategy to penalize the pixels with large representation errors. In [73], Maximum correntropy criterion (CESR) was proposed for robust face recognition. By half-quadratic optimization, CESR can be finally converted to an iterative reweighted problem. As there are also expression, pose, illumination and other unpredictable variations in the wild environment, more robust models should be developed to deal with complex variations.

Kernel representation Kernel trick is often used to map the samples in the original non-linear separable feature subspace to a high-dimensional feature space, in which features of the type are easily grouped together and the samples becomes linearly separable. Many linear classifiers and algorithms, e.g., SVM, PCA and LDA, are extended to kernel version, that is, kernel SVM, kernel PCA and kernel LDA. For representation based classifiers, by kernel mapping to the reproducing kernel Hilbert space (RKHS), kernel sparse representation [55, 56]and kernel collaborative representation [216, 219, 223, 232] are proposed respectively. For kernel representation based classifiers, representation residual can be used for classification [55, 216, 219, 223, 232]. Besides, kernel representation can also be combined with spatial pyramid matching. Instead of vector quantization [103], local sparse coding [212] or soft-threshold [34], kernel representation can also be used for coding process [55]. From the solution stability of linear system, in RKHS the linear system tends to be over-determined. Hence, the regularization seems less important for representation based classifiers [232].

Generic representation For representation based classifiers, we can seek a good representation with a over-complete dictionary. However, sometimes, the training samples are insufficient or the dictionary is not well trained. For example, in face recognition, the variations in the query face image is not contained in the training samples. In SRC, to deal with the noise in the query face image, an identity matrix is introduced to simulate the noise part [201]. To introduce more variations of face images, a generic dictionary that contains different face variations is introduced and an extended SRC (ESRC) is proposed [43] to solve small sample size problem in face recognition. In ESRC, the variation dictionary is obtained by the differences between a variation subset and a reference subset. Hence, ESRC can not contain all the possible face variations. In [210], a sparse variation dictionary learning (SVDL) method is proposed to learn a variation dictionary and use it for face recognition with single sample per person.

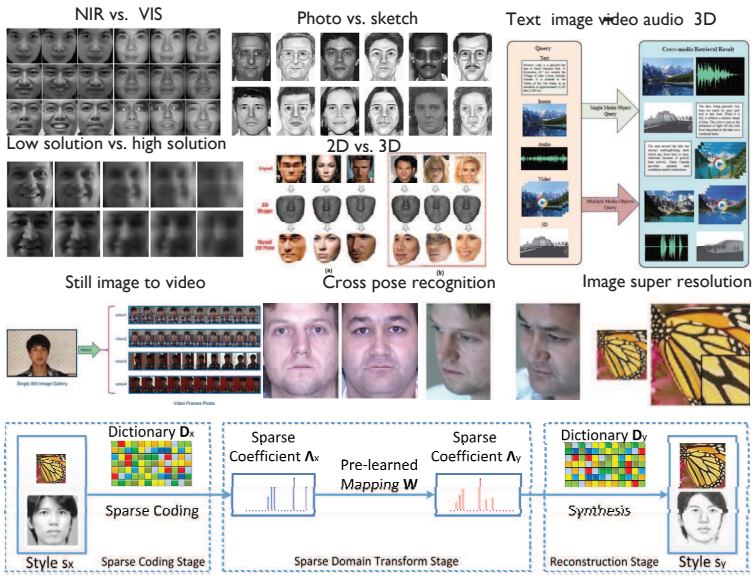
Multi-modal representation

With the rapid development of sensor techniques and widespread use of Internet, the diversity of data sources, data types and representations leads to an explosion of multi-modal data. Multi-modal data widely exists and is applied in biometrics, computer vision, multimedia, fault diagonal, remote sensing data, medical analysis, etc. Researches on human brain mechanism show that human beings can effectively store, transform and integrate the information from different sense organs. It becomes extremely important to investigate how to simulate the data processing mechanism of human beings to fuse multi-modal information for detection, recognition and prediction. The present multi-modal classification models can be

categorized into three types: feature-level modeling, decision-level modeling and deep learning. Feature-level modeling explores how to combine, project or transform features from different modalities, including feature stacking, multi-projections learning [26] and multi-modal dictionary learning [129]. Decision-level modeling aims to fuse multi-modal outputs or learn multi-modal classifiers, e.g., ensemble learning, multi-metric learning [140], multi-kernel learning [7] and multi-modal representation [219, 223]. Deep learning simulates the neural networks of human beings. It can separately or simultaneously conduct feature-level and decision-level modeling [131]. For multi-modal classification, the representation is jointly conducted with group sparsity regularization [223] or other smooth regularization.

Cross-modal representation

As shown in Fig. 1.3, different from multi-modal classification tasks, cross modal classification tasks need to match the object of one modality with the object of the other modality. There are quite a lot of cross modality classification tasks, e.g., photo-sketch face recognition [228], text to image retrieval [202], image to video face recognition [83], etc. To match objects of different modalities, distance metric learning, joint representation (regression) and deep learning methods have been proposed in the past few years. For cross modal representation models, the key motivation is that representation is conducted on each modality and a projection matrix is learned to connect the representation coefficients of different modalities [71, 80, 194, 222].



[H]

Figure 1.3 Cross-modal classification tasks.

1.2.2 Dictionary learning

For representation based models, one of the most important factors is the dictionary, on which a query signal is reconstructed. How to a design a good dictionary can track back to the hand-crafted bases, e.g., discrete cosine transform [4], wavelets [119], wedgelets [44], etc. Compared with these specially designed bases, learned dictionary aims minimize the reconstruction error and at the same time preserve the hidden structure or information within the data. The existing dictionary learning methods can be categorized into reconstructive and discriminative methods. Reconstructive methods emphasize the reconstruction ability of the dictionary, e.g., KSVD [3], method of optimal direction (MOD) [46]. Discriminative methods aim to introduce discrimination ability to representation residual or coding vectors. Instead of learning a dictionary for all the classes, class-specific dictionar-

ies are learned to introduce discrimination ability to the reconstruction residuals [57, 147, 215]. Intra-class Particularity and inter-class commonality are usually taken into account in the model of dictionary learning [57, 99]. In [147], structured incoherence is introduced to enhance the independence of the sub-dictionaries related with different classes. Another type is to learning discriminative coding vectors by dictionary learning. In [227], KSVD is extended to discriminative KSVD by simultaneously learning a dictionary and a linear classifier. In [90, 91], by introducing a lable-consistent item, the discrimination ability of coding vectors is enhanced. In [118], coding vectors are embedded into a logistic regression function and a task-driven dictionary learning method is proposed. In [215], class-specific dictionaries are learned while fish discrimination fidelity is imposed on coding vectors. The key challenge of dictionary learning for classification is to pursue the balance between representation and discrimination ability. Besides, the efficiency of dictionary learning is quite important, especially for some real-time applications, e.g, image retrieval and visual tracking.

1.3 Problems

Although representation based classification models have achieved great success in different classification tasks, there are still many problems with representation based models. When there are only a small number of training samples in the dictionary, the dictionary tends to be over-determined and therefore the query sample can not be well represented. With the development of sensors and digital devices, the data are consistently increasing with a high speed. When the number of the

training samples is very large, how can we deal with the storage burden and time complexity of representation based classifiers? Besides, the existing representation based classifiers can only be applied to single image based classification tasks. However, for video based face recognition and multi-view object recognition, the task becomes an image set classification problem. It is still an open problem to extend representation based classifiers from image based to image set based models. Finally, for representation based classifiers, the representation process is usually discriminative. Similar to discriminative classifiers (e.g., SVM), discriminative representation can be learned to enhance classification. In the following part, we will discuss the three problem mentioned above in details.

1.3.1 Representation with small sample size

In classification tasks, sometimes the available training samples are quite limited. This is called small sample size problem in machine learning. In face recognition, we have to deal with small sample size problems. Face recognition (FR) is a very active topic in computer vision research because of its wide range of applications, including access control, video surveillance, social network, photo management, criminal investigation, etc [86]. Though FR has been studied for many years, it is still a challenging task due to the many types of large face variations, e.g., pose, expressions, illuminations, corruption, occlusion and disguises. Furthermore, in applications such as smart cards, law enforcement, etc., we may have only one template sample of each subject, resulting in the single sample per person (SSPP) problem [175]. SSPP makes FR much more difficult because we have little information from the gallery set to predict the variations in the query face image [220].

Since the intra-class variations cannot be well estimated in the SSPP problem, the traditional discriminative subspace learning based FR methods can fail to work. In addition, since the number of samples per class is so small, the robustness of extracted features and the generalization ability of learned classifiers can be much reduced. For representation based classification models, the query face image can not be well reconstructed by the training images. Besides, as the number of sampler per class is quite small, the linear system seems to be over-determined. Hence, the solution is unstable and leads to misclassification.

1.3.2 Representation with big sample size

With the data rapidly increasing, there are large amounts of training samples and therefore the large-scale classification task is yielded. In this case, for representation based classifiers, the linear system tends to be over-complete. However, the massive training samples lead to large computation burden and high time complexity. Then how can we develop a representation based classifier with low computation burden and time complexity for large-scale tasks?

As shown in Fig. 1.4 the existing representation based classifiers all reply upon sample-level representation, i.e., a query sample can be linearly reconstructed by a set of sample bases. In nature, self-similarity widely exists, i.e., a part of an object is similar to other parts of itself, e.g., coastlines [120], stock market movements [19] and images [17]. Taking images for example, patches at different locations in an image perhaps are similar to each other, which is called non-local self-similarity. In image processing, the so-called non-local self-similarity has been successfully used in high performance image restoration and denoising [17]. As shown in Fig.

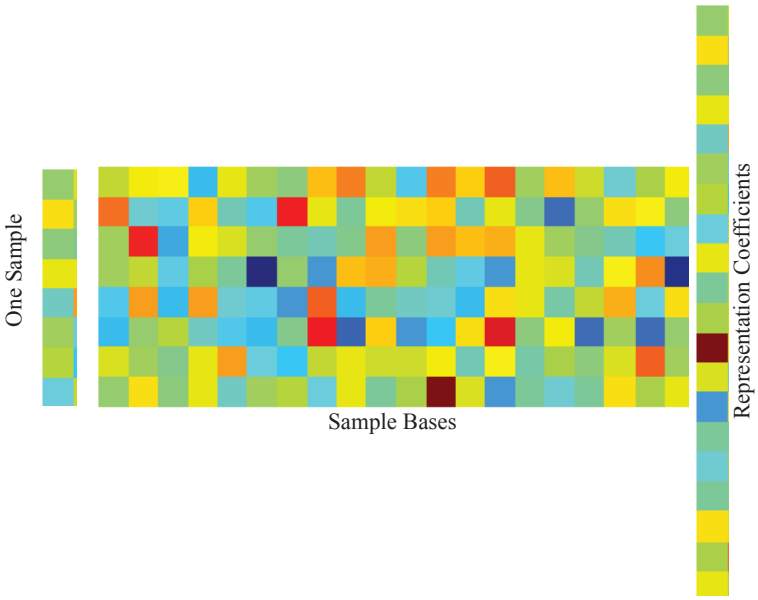


Figure 1.4 Sample-level representation.

1.5, one feature can be represented by its relevant features. Based on feature-level representation, a series of interesting models can be developed. The relationship between sample-level and feature-level relationship can also be further investigated.

1.3.3 Image set representation

Image set based classification has become increasingly important in face recognition [5, 21, 29, 40, 78, 136, 145, 193, 199, 206] and object categorization [98, 190] in recent years. Due to the rapid development of digital imaging and communication techniques, image sets can be easily collected from multi-view images using multiple cameras [98], long term observations [199], personal albums and news pictures [162], etc. Since the gallery image sets contain more within-class variations

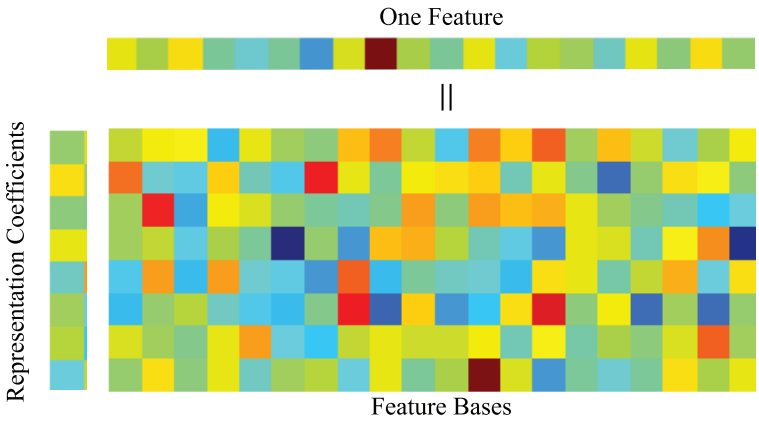


Figure 1.5 Feature-level representation.

of the subject [78], image set based face recognition (ISFR) has shown superior performance to single image based face recognition. One special case of ISFR is video based face recognition, which collects face image sets from consecutive video sequences [105, 171, 206]. As shown in Fig. 1.6, a query face image set is cropped from the query video and similarly the training face image sets are collected from the gallery videos. Then the face recognition problem becomes matching one image set with a set of training image sets.

One may apply SRC/CRC to ISFR by representing each image of the query set over all the gallery sets, and then using the average or minimal representation residual of the query set images for classification. However, such a scheme does not exploit the correlation and distinctiveness of sample images in the query set. If the average representation residual is used for classification, the discrimination of representation residuals by different classes will be reduced; if the minimal representation residual is used, the classification can suffer from the outlier images in the query set. In addition, there are redundancies in an image set. The redundancies

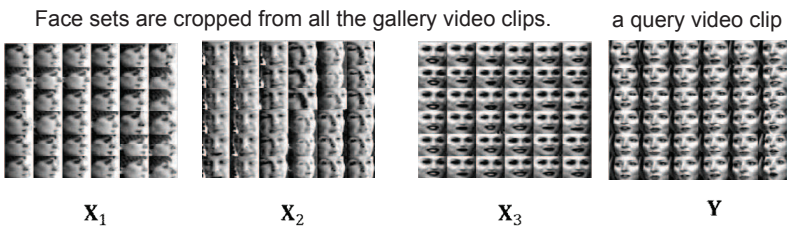


Figure 1.6 Image set based face recognition.

will lead to great storage burden and computational complexity, and deteriorate the recognition performance.

1.3.4 Discriminative representation

Representation based classifiers aim to well reconstruct a query sample by the given training samples or a specially learned dictionary. Then representation based residuals or coefficients are used for classification. However, the representation process is unsupervised and therefore lacks discrimination ability. Discriminative dictionary learning methods have been proposed to make the representation discriminative, e.g., FDDL [215], D-KSVD [227], etc. Besides, discriminative projections can also be learned to project the query sample and the dictionary to a low-dimensional discriminative feature space [132, 210]. Actually, representation based classifiers, e.g., nearest subspace classifier, can be considered as point to set distance based classifiers. Hence, learning a discriminative point-to-set distance can enhance the performance of representation based classifiers. Similarly, a set-to-set distance can also be learned in image set based classification tasks.

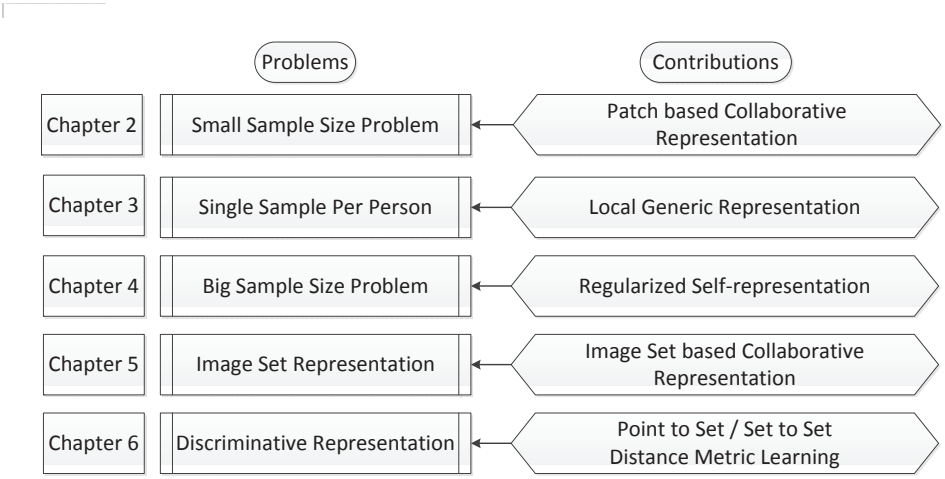


Figure 1.7 The main contributions of the thesis.

1.4 Thesis contributions

As shown in Fig. 1.7, the main contributions of this thesis are listed as follows:

In Chapter 2, to solve the over-determined representation problem in face recognition, we proposed a patch based CRC (PCRC) method and consequently the multi-scale version of it, i.e., MCPCRC, by margin distribution optimization. The query image was partitioned into a set of overlapped patches and each patch is collaboratively represented over the corresponding set of patches of all training samples. The classification outputs of all patches were then combined by voting. However, the patch size will have a great impact on the final classification result of PCRC. Therefore, we proposed to use multiple patch sizes and then optimally combine the multi-scale outputs by margin distribution optimization with l_1-norm regularization.

In Chapter 3, we proposed a local generic representation (LGR) based approach

for the challenging task of face recognition with single sample per person (SSPP). LGR utilizes the advantages of both patch based local representation and generic learning. A generic intra-class variation dictionary was constructed from a generic dataset, and it can well compensate for the face variations lacked in the SSPP gallery set. A patch gallery dictionary was built by using the gallery samples, which can more accurately represent the different parts of face images. Considering that the distribution of representation residual of different patches is highly non-Gaussian, a correntropy based metric was adopted to measure the loss of each patch so that the importance of different patches in face recognition can be more robustly evaluated. As a result, LGR can adaptively suppress the role of patches with large variations. The extensive experimental results on four benchmark face databases showed that LGR always achieves higher face recognition rate than the state-of-the-art SSPP methods used in competition.

In Chapter 4, for large-scale representation, we investigated the representation based classification problem from a "feature oriented" perspective. Different from the existing representation based classifiers that represent a sample as the linear combination of other samples, we explored to represent a feature by its relevant features in the data, which we call self-representation. A self-representation induced classifier (SRIC) was then proposed, which learns a self-representation matrix per class and uses these matrices for classification. The query sample is then classified to the class with the minimal reconstruction error. We proved that SRIC is equivalent to nearest subspace classifier (NSC) with l_2-norm regularization in terms of classification decision. Furthermore, it can be shown that SRIC is essentially the principle component analysis (PCA) with eigenvalue shrinkage. We then proposed

a discriminative SRIC (DSRIC) classifier, which not only minimizes the feature self-representation residual of this class but represents little the features of other classes. As the time complexity of SRIC and DSRIC is only related with the feature dimension, the proposed method can apply to the classification tasks with large number of training samples.

In Chapter 5, for image set based representation, we proposed a novel image set based collaborative representation and classification (ISCRC) scheme for image set based face recognition (ISFR). The query set was modeled as a convex or regularized hull, and a collaborative representation based set to sets distance (CRSSD) was defined by representing the hull of query set over all the gallery sets. The CRSSD considers the correlation and distinction of sample images within the query set and the relationship between the gallery sets. With CRSSD, the representation residual of the hull of query set by each gallery set can be computed and used for classification.

In Chapter 6, we extended the point-to-point distance metric learning to point-to-set distance metric learning (PSDML) and set-to-set distance metric learning (SSDML). Positive and negative sample pairs were generated from training sample sets by computing point-to-set distance (PSD) and set-to-set distance (SSD). Each sample pair was represented by its covariance matrix and a covariance kernel based discrimination function was proposed for sample pair classification. Finally, we showed that the proposed metric learning problem can be efficiently solved by SVM solvers. Experiments on various visual classification problems demonstrated that the proposed PSDML and SSDML methods can effectively improve the performance of PSD and SSD based classification. Compared with the state-of-the-art

metric learning methods such as LMNN, ITML and MCML, the proposed method can achieve better classification accuracy and is significantly faster in training.

Chapter 2

Patch based Collaborative Representation

In computer vision and pattern recognition tasks, the acquisition of training samples is sometimes quite difficult and therefore results in small sample size (SSS) classification problem, especially in face recognition. Collaborative representation represents a query sample on a specially designed or learned dictionary and then use the representation residual for classification. Unfortunately, representation based classifiers may fail for SSS problems in that the representation can be inaccurate and the linear system tends to be over-determined. In this chapter, we investigate the SSS problems in face recognition from the perspective of patch based collaborative representation.

2.1 Introduction

Face recognition (FR) has been an active research topic in computer vision and pattern recognition for many years [229]. In spite of the tremendous achievements, there are still many challenges caused by the large face appearance variations of illumination, expression, pose, noise, occlusion, etc [144]. Particularly, the small sample size problem is one of the most fundamental and challenging issues in FR. In many real-world applications such as smart cards, law enforcement, surveillance and access control, the training samples of many subjects are often very limited [175]. Unfortunately, the performance of appearance based FR methods, such as the classical Eigenface [225], Fisherface [11], LPP [75] and the variants of them [207], degrades much with the decrease of training samples.

As a generalization and extension of the nearest neighbor, nearest line, nearest plane and nearest subspace classifiers, the sparse representation based classification (SRC) [201] scheme shows very interesting FR results. SRC represents a query face as a sparse linear combination of the training samples from all classes, and classifies it to the class which has the least representation residual. However, in [226] it was indicated that the costly l_1-norm sparse regularization on the representation vector in SRC is not necessary, and l_2-norm regularization can lead to similar FR results but with much lower computational cost. The collaborative representation based classification (CRC) was then proposed in [226] by representing the query sample with non-sparse l_2-regularization. However, both CRC and SRC suffer serious performance degradation when the training sample size is very small and hence the query sample cannot be well represented [200].

To solve the SSS problem, virtual samples and generic training set were used in

[173]. On the other hand, the trained classifiers will become unstable and have poor generalization ability when the available samples are insufficient, and hence ensemble learning has been widely applied to FR and has led to significant improvement in recognition rate and robustness [174][109][102]. These methods can be roughly divided into three categories. The first category of methods is patch (or block) based methods, which usually involve steps of local region partition, local feature extraction and classification combination [101][102]. The recognition rate of patch based methods is much affected by patch size, which is often set by experimental experience [27] [174]. Considering that the global and local features can provide complementary information, the second category of methods combines the global and local features for classification [109][172]. Third, a very popular category of methods uses multiple feature extractors to extract different types of facial features, and then uses classifier fusion for classification. For example, in [198][66], local features such as SIFT, LBP, Gabor response and gray values are combined for face verification.

Human faces exhibit distinct structures and characteristics when observed on different scales [109]. Combining the information on different scales could not only lead to much FR improvement but also provide us a simple and effective way for scale-insensitive models. How to combine multi-scale information is essentially an ensemble learning task. AdaBoost [155] is one of the most successful ensemble learning techniques due to its excellent performance and broad applications in face and object detection, visual tracking, etc. The success of AdaBoost actually attributes to margin distribution optimization [151][168][169], and AdaBoost approximately minimizes the loss criterion with l_1-regularization on the coefficient

vector [155]. In [166], Shawe-Taylor gave the bound of AdaBoost's generalization error based on margin distribution, which shows that the loss of margin and the norm of coefficient vector could be minimized.

In this chapter, to improve the performance of CRC in SSS problem, we propose to conduct CRC on patches, and the so-called patch based CRC (PCRC) classifies the query sample by combining the recognition outputs of all the overlapped patches, each of which is collaboratively represented by the corresponding patches of training samples. Similar to those patch based methods, PCRC is a patch size sensitive method, while the optimal patch size varies with training sample size and databases. In order for a patch size robust scheme, we then propose a multi-scale PCRC (MSPCRC) method by combining the information on different scales. MSPCRC considers PCRC on each scale as a base classifier and learns scale weights to fuse multi-scale decisions. Scale weights are learned by minimizing the square loss of margin, and sparse l_1-norm regularization is imposed on the weights to get better margin distribution.

The rest of this chapter is organized as follows. Section 2 describes PCRC. Section 3 presents the margin distribution optimization for multi-scale ensemble. Section 4 conducts experiments and conclusions are made in Section 5.

2.2 Patch based CRC

In [226], Zhang et al. proposed to use the regularized least square model for collaborative representation based classification (CRC) of face images. Given a set of training samples, denote by $\boldsymbol{X}_k \in \mathfrak{R}^{m\times n_k}$ the dataset of the k^{th} class, and each col-

umn of $\boldsymbol{X}_k$ is a sample of class k. Suppose that we have c classes of subjects, and let $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_c]$. Given a query sample $\boldsymbol{y}$, the collaborative representation of it is

$$\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}}\{||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{a}||_2^2 + \lambda\,||\boldsymbol{a}||_2^2\} \tag{2.1}$$

The solution of CRC is $\hat{\boldsymbol{a}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda \cdot \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$. The classification of CRC is performed by checking which class yields the minimal regularized reconstruction error. The recognition output of the query sample $\boldsymbol{y}$ is Indentity($\boldsymbol{y}$) = argmin$_k\{r_k\}$, where $r_k = ||\boldsymbol{y} - \boldsymbol{X}_k \cdot \hat{\boldsymbol{a}}_k||_2/||\hat{\boldsymbol{a}}_k||_2$ and $\hat{\boldsymbol{a}} = [\hat{\boldsymbol{a}}_1; \hat{\boldsymbol{a}}_2; ...; \hat{\boldsymbol{a}}_c]$.

When the linear system determined by dictionary $\boldsymbol{X}$ is under-determined, the linear representation of the query sample over $\boldsymbol{X}$ can be very accurate while regularization on $\boldsymbol{a}$ is necessary for a unique and stable solution [200]. Once the available samples per subject are very limited, CRC may fail because the linear representation of the query sample $\boldsymbol{y}$ may not be accurate. To alleviate this problem, patch based CRC (PCRC) can be introduced. As shown in Fig. 2.1, the query image $\boldsymbol{y}$ is firstly divided into a set of overlapped block patches $\{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_q\}$. Then each patch $\boldsymbol{y}_j$ is represented over local dictionary $\boldsymbol{M}_j$, which is extracted from $\boldsymbol{X}$ at the corresponding location to patch $\boldsymbol{y}_j$. Since the linear system determined by local dictionary $\boldsymbol{M}_j$ tends to be under-determined, the patch based representation is more accurate than the whole image based representation. Finally, plurality or linear weighted combination can be applied to the many patch based recognition outputs for a final classification.

For each local patch, the local features such as LBP and Gabor features can be used in PCRC. Considering that the focus of this chapter is to validate the effectiveness of PCRC strategy instead of local features, for simplicity and clarity the raw

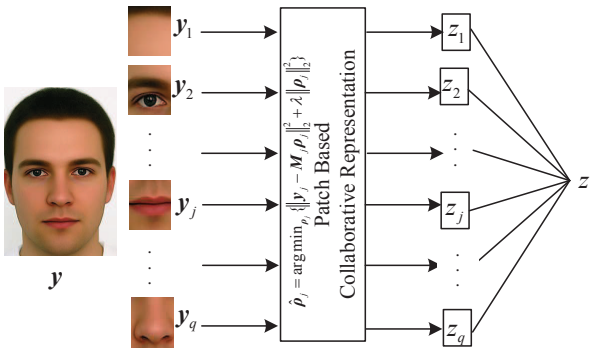


Figure 2.1 Diagram of patch based collaborative representation for face classification.

gray value features in each patch are used. For patch $\boldsymbol{y}_j$, its representation over $\boldsymbol{M}_j$ is obtained by

$$\hat{\boldsymbol{\rho}}_j = \arg\min_{\boldsymbol{\rho}_j} \{\left\|\boldsymbol{y}_j - \boldsymbol{M}_j\boldsymbol{\rho}_j\right\|_2^2 + \lambda \left\|\boldsymbol{\rho}_j\right\|_2^2\} \tag{2.2}$$

$\boldsymbol{M}_j$ is a local dictionary. Denote by $\boldsymbol{M}_{jk}$ the sub-dictionary of the k^{th} class, and each column of $\boldsymbol{M}_{jk}$ is a patch of class k. Then $\boldsymbol{M}_j = [\boldsymbol{M}_{j1}, \boldsymbol{M}_{j2}, ..., \boldsymbol{M}_{jc}]$. The recognition output z_j of patch $\boldsymbol{y}_j$ is Identity($\boldsymbol{y}_j$)= argmin$_k\{r_{jk}\}$, where $r_{jk} = \left\|\boldsymbol{y}_j - \boldsymbol{M}_{jk} \cdot \hat{\boldsymbol{\rho}}_{jk}\right\|_2 / \left\|\hat{\boldsymbol{\rho}}_{jk}\right\|_2$ and $\hat{\boldsymbol{\rho}}_j = [\hat{\boldsymbol{\rho}}_{j1}; \hat{\boldsymbol{\rho}}_{j2}; ...; \hat{\boldsymbol{\rho}}_{jc}]$.

The classification outputs of all patches can then be combined. Majority voting [101], linear weighted combination [174], kernel plurality [102] and probabilistic model [109] can be employed for the combination. As shown in [101] and [172], the weighted combination leads to little improvement compared to the simple majority voting. Hence, we use the majority voting for the final decision making.

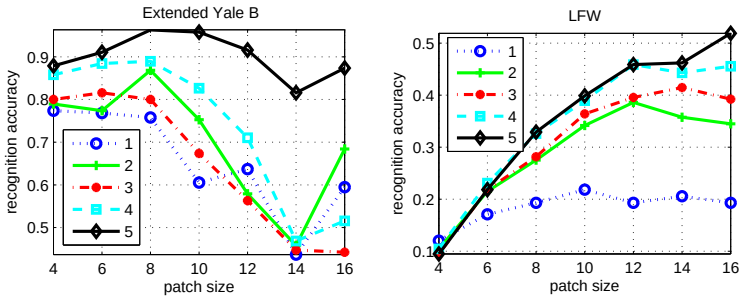


Figure 2.2 Impact of patch size on PCRC (1-5 represent the training sample size per subject).

2.3 Multi-scale ensemble

In the proposed PCRC, the patch size, or we call it the patch scale in this chapter, will have a great impact on the recognition performance and it is not a trivial work to pre-define an optimal scale for a database. Fig. 2.2 shows the FR accuracy under different patch sizes and training sample sizes on the Extended Yale B and LFW databases. One can have the following observations. First, the optimal scale varies with the number of training samples per subject. Second, for different databases, the optimal scale also varies a lot. This difficulty can be solved by fusing the multi-scale PCRC results adaptively, via which we can not only be free of the scale selection problem but also exploit the complementary information across scales to improve the FR accuracy and robustness. To this end, we propose an ensemble learning method to combine multi-scale information optimally.

The flowchart of the proposed method is given in Fig. 2.3. On different scales with various patch sizes, we can get the recognition outputs by PCRC. We then find a set of optimal weight $\boldsymbol{w}$ to fuse the outputs. In this chapter, we propose to learn $\boldsymbol{w}$ from the training samples by optimizing margin distribution.

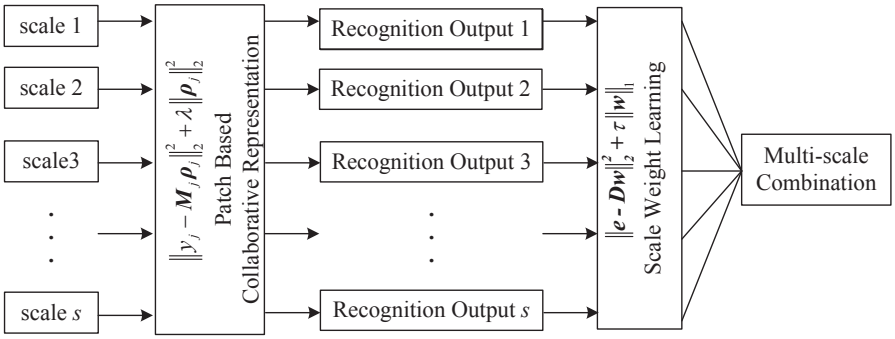


Figure 2.3 Flow chart of multi-scale learning for PCRC.

2.3.1 The objective function for ensemble optimization

The multi-scale ensemble of PCRC outputs can be considered as a special classification task. Suppose there are two scales and two classes labeled as +1 and -1. For a given sample, on each scale we can have a classification output, +1 or -1, and thus the classification output on the two scales of each sample has four possible situations, as shown as the four vertexes in Fig. 2.4(a). Given a set of training samples, we aim to find a classification line $f = sgn(w_1 z_1 + w_2 z_2)$ that crosses the origin to make all the given samples correctly classified, where z_1 and z_2 represent the classification outputs on the two scales and w_1 and w_2 represent the weights. As to the task in Fig. 2.4(a), if samples on vertexes $\{A_2, A_4\}$ belong to the first class (+1) and samples on vertexes $\{A_1, A_3\}$ belong to the second class (-1), there are several classification lines that can correctly classify all the samples. Similar to feature selection [60], the importance of one scale is proportional to the weight value assigned to it.

For binary classification problems, given a set of samples $\boldsymbol{S} = \{(\boldsymbol{x}_i, z_i)\}, i = 1, 2, ...n$, $z_i \in \{+1, -1\}$ and s scales, the recognition results on s different scales form a space $\boldsymbol{H} \in \Re^{n \times s}$. Let $\boldsymbol{w} = [w_1, w_2, ..., w_s]$ be the scale weight vector and

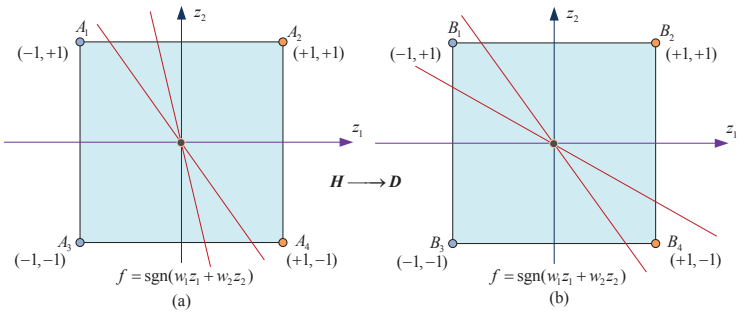


Figure 2.4 Illustration of the multi-scale ensemble learning problem.

$\sum_{j=1}^{s} w_j = 1$.

Definition Given a sample $\boldsymbol{x}_i \in \boldsymbol{S}$, the recognition outputs on s different scales are $\{h_{ij}\}$, $j = 1, 2, ..., s$. The discriminant function is $f = \mathrm{sgn}(\sum_{j=1}^{s} w_j h_{ij})$. The margin of sample $\boldsymbol{x}_i$ can be defined as [160]:

$$\varepsilon(\boldsymbol{x}_i) = z_i \sum_{j=1}^{s} w_j h_{ij} \tag{2.3}$$

Obviously, if $\varepsilon(\boldsymbol{x}_i) > 0$, then sample $\boldsymbol{x}_i \in \boldsymbol{S}$ is correctly classified; if $\varepsilon(\boldsymbol{x}_i) < 0$, then sample $\boldsymbol{x}_i \in \boldsymbol{S}$ is misclassified; if $\varepsilon(\boldsymbol{x}_i) = 0$, we cannot decide the label of sample $\boldsymbol{x}_i$. It is similar to linear classifiers (e.g., LSVM). Since Definition 1 is only suitable for binary classification, we define the following decision matrix in order for multi-class classification tasks.

Definition As to multi-class classification, given a sample $\boldsymbol{x}_i \in \boldsymbol{S}$, the recognition outputs on s different scales are $\{h_{ij}\}$, $j = 1, 2, ..., s$. The decision matrix $\boldsymbol{D} = \left\{d_{ij}\right\}, i = 1, 2, ..., n, j = 1, 2, ..., s$, is defined as:

$$d_{ij} = g(z_i, h_{ij}) = \begin{cases} +1, \text{if } z_i = h_{ij} \\ -1, \text{if } z_i \neq h_{ij} \end{cases} \tag{2.4}$$

where z_i is the label of sample $\boldsymbol{x}_i$.

Clearly, $d_{ij} = +1$ means that $\boldsymbol{x}_i$ is correctly classified on the j^{th} scale. Otherwise, it is misclassified.

Definition Given a sample $\boldsymbol{x}_i \in \boldsymbol{S}$, the classification outputs on s different scales are $\{h_{ij}\}$, $j = 1, 2, ..., s$. The ensemble margin of $\boldsymbol{x}_i \in \boldsymbol{S}$ can be defined as:

$$\varepsilon(\boldsymbol{x}_i) = \sum_{j=1}^{s} w_j d_{ij} \tag{2.5}$$

Ensemble margin reflects the misclassification degree in classifier fusion. Samples with positive margin are correctly classified. As shown in Fig. 2.4(b), +1 and -1 represent the elements in the decision matrix $\boldsymbol{D}$, and then the margin of samples on vertex B_2 is 1 (i.e., correctly classified on all scales) , while the margin of samples on vertex B_3 is -1 (i.e., misclassified on all scales). The margin of samples on vertexs B_1 and B_4 is between -1 and +1. In this case, how should we choose the scale weights to get better combination result? We should make the ensemble margin as larger as possible by scale weight learning. Margin maximization is usually converted into a loss minimization problem [183][155][168].

If the ensemble margin of a sample $\boldsymbol{x}_i$ is $\varepsilon(\boldsymbol{x}_i)$, then the ensemble loss of sample $\boldsymbol{x}_i$ is

$$l_{x_i} = l(\varepsilon(x_i)) = l(\sum_{j=1}^{s} w_j d_{ij}) \tag{2.6}$$

We adopt the square loss used in CRC [226], SRC [201], LS-SVM [183] and least square regression [148]. For a sample set $\boldsymbol{S}$, the ensemble square loss is

$$\begin{aligned} l(\boldsymbol{S}) &= \sum_{i=1}^{n} l_{x_i} = \sum_{i=1}^{n} [1 - \varepsilon(x_i)]^2 \\ &= \sum_{i=1}^{n} [1 - \sum_{j=1}^{s} w_j d_{ij}]^2 = \|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w}\|_2^2 \end{aligned} \tag{2.7}$$

where $\boldsymbol{e}$ is a vector whose elements are 1 and length is s.

2.3.2 Constrained l_1-regularized optimization

To learn the optimal scale weights, we should minimize the ensemble loss in Eq. (7). However, there may be many solutions that can minimize the loss for the given task, as illustrated in Fig. 2.4. Clearly, we should regularize the objective function in Eq. (7) in order for a unique and robust solution. In [155], Saharon et al. showed that AdaBoost approximately minimizes its loss criterion with l_1-regularization imposed on the coefficient vector. In [169], it was shown that AdaBoost optimizes margin distribution rather than minimum margin. Shawe-Taylor gave the bound on generalization error based on margin distribution for linear classifiers ($f = \boldsymbol{w}\boldsymbol{x} + b$) and showed that both the square loss (when $\sum_{j=1}^{s} w_j = 1$ and $x \in \{+1, -1\}$) and the norm of $\boldsymbol{w}$ should be minimized to improve the generalization ability [166].

Inspired by the principle of AdaBoost, we propose the following constrained l_1-regularized least square optimization to minimize the ensemble loss and solve the weights:

$$\begin{aligned} \hat{\boldsymbol{w}} &= \arg\min_{\boldsymbol{w}}\{\|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \tau\|\boldsymbol{w}\|_1\} \\ &s.t. \sum_{j=1}^{s} w_j = 1, w_j > 0, j = 1, 2, ..., s \end{aligned} \tag{2.8}$$

where τ is the regularization parameter.

For the constraint $\sum_{j=1}^{s} w_j = 1$, it equals to $\boldsymbol{e}\boldsymbol{w} = 1$, where $\boldsymbol{e} = [1; 1; ...; 1]$ is a column vector, and then

$$\|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w}\|_2^2 = \|\boldsymbol{e} - \boldsymbol{D}\boldsymbol{w} + 1 - \boldsymbol{e}\boldsymbol{w}\|_2^2 = \|[\boldsymbol{e}; 1] - [\boldsymbol{D}, \boldsymbol{e}]\boldsymbol{w}\|_2^2 \tag{2.9}$$

Let $\hat{\boldsymbol{e}} = [\boldsymbol{e}; 1]$, $\hat{\boldsymbol{D}} = [\boldsymbol{D}, \boldsymbol{e}]$, then we can get

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}}\{\left\|\hat{\boldsymbol{e}} - \hat{\boldsymbol{D}}\boldsymbol{w}\right\|_2^2 + \tau\|\boldsymbol{w}\|_1\} \; s.t. \; w_j > 0, \; j = 1, 2, ..., s \tag{2.10}$$

Since the size of the decision matrix is very small (e.g., the size of decision matrix for the LFW database is 632×7 when the training sample size per subject is 5 and 7 scales are selected), $\boldsymbol{w}$ can be easily solved by some representative l_1-minimization approaches [208]. In this chapter l_1_ls is used for its accuracy and stable solution [96]. The proposed ensemble learning algorithm for multi-scale PCRC (MSPCRC) is summarized in Table 2.1. After scale weight learning, for a query sample $\boldsymbol{x}_i$, the recognition output is $z_i = \arg\max_k\{\sum w_j | h_{ij} = k\}$.

It should be noted that though the form of multi-scale ensemble in Eq. (10) is similar to the step of coding in CRC (Eq. (1)) and SRC, their physical meanings are different. The square loss in CRC and SRC is the reconstruction error while in multi-scale ensemble learning the square loss is the function of classification margin. The l_1-norm regularization used in SRC is to sparsify the coding coefficient to enhance classification accuracy, while the l_1-norm regularization used in multi-scale ensemble learning is to suppress the effect of less-useful scales.

Table 2.1 The algorithm of multi-scale ensemble learning for PCRC.

1: Choose s patch sizes $\delta = \{\delta_1, \delta_2, ..., \delta_s\}$

2: Get recognition outputs $\{h_{ij}\}$ by PCRC

3: Get the decision matrix

$$d_{ij} = g(z_i, h_{ij}) = \begin{cases} +1, if\ z_i = h_{ij} \\ -1, if\ z_i \neq h_{ij} \end{cases}$$

4: Learn scale weights

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \left\| \hat{\boldsymbol{e}} - \hat{\boldsymbol{D}}\boldsymbol{w} \right\|_2^2 + \tau \|\boldsymbol{w}\|_1 \ \ s.t.\ w_j > 0,\ j = 1, 2, ..., s$$

2.4 Experimental analysis

We use the Extended Yale B [59], Multi-PIE [64] and AR [123] databases in controlled environments together with the LFW database [82] in uncontrolled environments to test the FR performance of the proposed method.

The baseline CRC, SRC and NN methods, and the state-of-the-art patch based methods including BlockFLD [27], Volterrafaces [101] and patch based nearest neighbor (PNN) classifier [102] are used for comparison. As the average accuracy improvement of kernel plurality [102] compared to vote is only about 1%, we report the result of PNN and Volterrafaces with majority voting. For Volterrafaces, the best recognition performance is reported with different kernel sizes and patch sizes. As linear kernel outperforms quadratic kernel on all the four databases, we only report the performance of liner kernel for Volterrafaces. For BlockFLD [27], the performance of CS2 (combine outputs of different blocks), which is better than CS1 (combine projected blocks as a feature), is reported.

In all the following experiments, the program is run for 20 times on each database and the average results are reported. Seven scales are used in our MSPCRC method and the patch sizes are 4×4, 6×6, 8×8, 10×10, 12×12, 14×14, 16×16. In single scale based PCRC and PNN, the patches are overlapped and the patch size is set as 10×10 (overlap is 5 pixels). The parameter λ used in SRC, CRC, PCRC and MSPCRC are set as 0.001, 0.005, 0.001 and 0.001, respectively. Parameter τ (Eq. (10)) is set as 0.1 for MSPCRC. For BlockFLD, we tried three different sizes (4×4, 8×8, 10×10 for 32×32 image and 10×10,15×15, 20×20 for 80×80 image) and report the result of the best size 8×8 (32×32 image) and 10×10 (80×80 image) for all the databases.

For scale weight learning, we divide the training set into subset1 (one image per individual is selected) and subset2 (the rest of the training set). Then samples from subset1 are classified by PCRC using subset2 as the training set on seven scales so that the weights can be learned. Obviously, as least two samples per subject are needed to learn the scale weights. Hence, we first test the performance of PCRC and MSPCRC with 2 to 5 training samples per subject. Then when there is only one sample per person, only the result of PCRC is reported.

2.4.1 Extended Yale B database

The Extended Yale B face database [59] contains 38 human subjects under 9 poses and 64 illumination conditions. All frontal-face images marked with P00 were used in our experiment. The face images are resized to 32×32. We randomly choose 2∼5 samples from the first 32 images for training and choose 5 samples from the other 32 images for test. The experimental results are shown in Table 6.9. It can be clearly seen that MSPCRC achieves the highest recognition rate on all experiments with the training sample size increasing from 2 to 5. Compared to PCRC, MSPCRC leads to much better results, validating the effectiveness of multi-scale ensemble learning.

2.4.2 Multi-PIE database

The Multi-PIE database [64] contains a total of more than 750,000 images from 337 individuals, captured under 15 viewpoints and 19 illumination conditions in four recording sessions. A subset that contains images of 164 subjects from session 3 is selected, and there are 10 images with neutral expression and 10 images with smile

Table 2.2 Recognition accuracy (%) on the extended Yale B database.

Method	2	3	4	5
CRC[226]	61.3±16.6	74.0±15.5	81.4±17.6	87.8±13.7
SRC[201]	64.2±17.2	74.2±15.2	82.6±16.8	89.0±12.5
NN	49.8±17.3	55.8±16.6	63.7±17.2	68.4±16.8
PNN[102]	60.8±14.4	65.6±15.1	73.8±15.8	79.7±14.6
BlockFLD[27]	79.5±8.4	83.8±7.8	88.3±5.4	90.7±5.5
Volterra[101]	69.8±12.9	79.5±12.3	84.0±9.6	86.4±9.6
PCRC	75.7±12.6	82.8±12.4	88.7±8.4	92.0±8.2
MSPCRC	**83.0±9.2**	**88.4±10.1**	**92.5±6.8**	**95.0±6.6**

expression per person. To make the FR problem more challenging, we randomly choose 2~5 samples per subject from images with neutral expression for training and randomly choose 3 samples from images with smile expression for test. The face images are resized to 32×32. The FR results are listed in Table 2.3. Similar to the results on the Extended Yale B database, PCRC and MSPCRC lead to much improvement in FR rate compared with the other methods. MSPCRC is always better than PCRC since it combines the multi-scale decisions.

2.4.3 AR database

The AR face database [123] contains over 4,000 color face images of 126 people, including frontal views of faces with different facial expressions, lighting conditions and occlusions. As in [226], a subset with only illumination and expression changes

Table 2.3 Recognition accuracy (%) on the Multi-PIE database.

Method	2	3	4	5
CRC[226]	62.6±13.8	74.3±6.3	78.5±5.2	80.4±3.7
SRC[201]	61.9±14.0	73.2±8.9	78.6±6.5	80.8±4.2
NN	54.9±14.5	64.7±12.1	71.9±9.9	74.5±8.8
PNN[102]	54.4±14.9	63.2±14.0	72.3±10.7	76.7±8.8
BlockFLD[27]	66.1±6.9	71.1±5.7	76.4±4.6	79.2±3.2
Volterra[101]	52.2±11.3	57.6±7.6	62.4±6.0	65.4±4.8
PCRC	68.8±10.9	76.0±6.2	79.4±4.8	81.3±3.7
MSPCRC	**72.4±10.5**	**79.6±5.9**	**83.6±4.0**	**84.6±2.6**

that contains 50 male subjects and 50 female subjects was chosen from the AR dataset in our experiments. For each subject, we randomly choose 2~5 samples from session 1 for training and choose 3 samples from session 2 for test. The face images are resized to 32×32.

The recognition accuracy on the AR database is shown in Table 2.4. The proposed methods show superior performance to all the other methods. Different from the results on the Extended Yale B and Multi-PIE databases, multi-scale ensemble learning in MSPCRC only leads to a little improvement over PCRC. That is because in this experiment the average weight value (over different training sample sizes) for scale 10×10 is about 0.9, which indicates that 10×10 is a very suitable patch size for PCRC in the AR database.

Table 2.4 Recognition accuracy (%) on the AR database.

Method	2	3	4	5
CRC[226]	69.9±12.6	80.6±10.4	83.8±9.6	89.1±6.2
SRC[201]	69.7±14.8	79.0±10.6	83.5±8.9	88.2±5.7
NN	48.5±9.5	54.7±9.0	58.5±9.1	63.2±7.0
PNN[102]	72.7±14.2	82.4±9.3	87.6±8.0	92.2±6.0
BlockFLD[27]	71.5±11.5	78.6±9.8	84.2±8.7	87.6±4.2
Volterra[101]	65.4±12.0	74.9±11.1	79.8±10.5	85.2±6.8
PCRC	82.2±11.3	87.7±9.4	89.9±8.5	92.9±6.7
MSPCRC	**82.3±11.5**	**87.8±10.5**	**90.2±9.1**	**93.6±7.6**

2.4.4 LFW database

The LFW database [82] contains images of 5,749 different individuals in unconstrained environment. LFW-a is a version of LFW after alignment using commercial face alignment software [197]. We gathered the subjects including no less than ten samples and then get a dataset with 158 subjects from LFW-a. For each subject, 2~5 samples are randomly chosen for training and another 2 samples for test. The images are firstly cropped to 121×121 and then resized to 32×32. The FR rates on the LFW dataset are listed in Table 2.5. One can see that PCRC and MSPCRC work much better than other methods, while the recognition performance is greatly improved by MSPCRC.

Table 2.5 Recognition accuracy (%) on the LFW database.

Method	2	3	4	5
CRC[226]	24.7±2.1	31.9±2.4	37.8±2.6	42.0±3.2
SRC[201]	24.4±2.4	32.7±3.2	38.7±2.4	44.1±2.6
NN	9.3±1.7	11.4±1.8	13.0±1.7	14.3±1.9
PNN[102]	23.1±2.4	28.1±3.1	33.2±3.1	37.4±2.7
BlockFLD[27]	18.0±2.1	22.3±2.1	26.2±2.6	28.4±2.5
Volterra[101]	26.0±3.0	32.0±3.4	36.4±3.3	40.3±2.7
PCRC	32.0±1.9	37.0±2.8	40.2±2.5	42.9±2.6
MSPCRC	**35.0±1.6**	**41.1±2.8**	**46.0±3.0**	**49.0±2.9**

2.4.5 Single sample per person (SSPP)

As there is only one sample per person, the proposed ensemble learning cannot be conducted. We report the recognition accuracy of PCRC on one scale for all the databases. The images are resized to 32×32 and 80×80, and the corresponding patch size is set as 8×8 and 20×20, respectively, for PCRC. When the image size is 80×80, the neighbor patches are used to construct the local dictionary. Since volterrafaces cannot deal with SSPP problem, its performance is not reported. BlockFLD (CS2) [27], AGL [173] and FLDA_single [54], which are methods specially designed for SSPP problem are compared. The results are listed in Table 2.6. The performance of PCRC is much better than SRC, CRC, NN, PNN, FLDA_single, and BlockFLD. Compared with AGL (adaptive generic learning) method, which uses an additional generic set to learn the projection matrix, the proposed PCRC

shows better performance on the MPIE, AR and LFW databases without using any additional information apart from the training set.

Table 2.6 Recognition accuracy (%) for SSPP.

32×32	Yale B	Multi-PIE	AR	LFW
CRC[226]	39.8±20.5	47.2±19.0	42.9±14.6	15.5±22.0
SRC[201]	38.7±20.5	48.2±18.6	44.9±14.8	14.7±1.9
NN	35.4±19.8	42.9±17.0	35.4±12.0	7.0±1.6
PNN[102]	45.1±18.3	40.1±17.7	54.4±19.5	15.8±2.0
BlockFLD[27]	63.1±15.0	56.9±9.7	52.1±19.8	11.8±1.4
FLDA_single[54]	39.9±21.4	43.5±14.8	37.2±10.4	6.7±1.5
AGL[173]	**75.9±12.2**	58.9±14.8	52.1±15.9	14.3±1.4
PCRC	66.5±16.3	**59.1±13.3**	**65.4±20.9**	**21.1±2.2**
80×80	Yale B	Multi-PIE	AR	LFW
CRC[226]	42.0±20.2	49.2±18.2	46.8±17.2	14.6±2.4
SRC[201]	39.3±19.6	48.3±16.7	42.0±13.3	12.6±1.8
NN	37.2±20.2	44.5±17.3	36.8±12.3	7.0±1.5
PNN[102]	57.9±18.6	49.1±17.3	61.0±19.3	16.0±2.3
BlockFLD[27]	65.7±13.3	51.9±5.6	41.9±17.8	4.9±1.3
FLDA_single[54]	41.2±20.9	39.3±10.5	32.9±12.0	8.7±1.8
AGL[173]	**79.1±12.7**	58.5±24.8	51.7±16.7	12.6±2.1
PCRC	76.7±17.4	**69.5±10.4**	**69.5±22.6**	**25.0±1.8**

2.5 Conclusions and discussions

In order for a more effective face recognition when the number of training samples per class is small, in this chapter we proposed a patch based CRC (PCRC) method and consequently the multi-scale version of it, i.e., MSPCRC, by margin distribution optimization. The query image was partitioned into a set of overlapped patches and each patch is collaboratively represented over the corresponding set of patches of all training samples. The classification outputs of all patches were then combined by voting. However, the patch size will have a great impact on the final classification result of PCRC. Therefore, we proposed to use multiple patch sizes and then optimally combine the multi-scale outputs by margin distribution optimization with l_1-norm regularization. Our experimental results on controlled and uncontrolled face databases showed that MSPCRC outperforms not only much the CRC and SRC benchmarks, but also state-of-the-art patch based methods such as BLDA and Volterrafaces, especially when the training samples size is very small.

For PCRC and MSPCRC, the projection matrices and scale-weights can be off-line learned. Hence, in the testing stage, PCRC is very fast besides its superior performance. As there is only one sample per class in face recognition with SSPP, the scale-weights can not be learned. Therefore, it is still an unsolved problem to learn a general scale-weights for PCRC using generic training set. Finally, although patch based representation tends to be more flexible for classification, the face variations in the gallery set still can not well represent the query face image. Hence, it is desirable to introduce more inter-class face variations to help representation.

Chapter 3

Local Generic Representation for Single Sample per Person

In Chapter 2, patch based collaborative representation is proposed to solve SSS problem. However, the variations in the gallery face images still cannot well represent the variation in the query sample. Considering the similarity of face images across subjects, a generic training set can be used to compensate for the shortage of samples in FR. Besides, the importance of different parts of faces varies. Hence, in this chapter, we take the advantage of the generic variation dictionary and consider the distinctiveness of different face parts to develop local generic representation based classifiers.

3.1 Introduction

Face recognition (FR) is a very active topic in computer vision research because of its wide range of applications, including access control, video surveillance, social network, photo management, criminal investigation, etc [86]. Though FR has been studied for many years, it is still a challenging task due to the many types of large face variations, e.g., pose, expressions, illuminations, corruption, occlusion and disguises. Furthermore, in applications such as smart cards, law enforcement, etc., we may have only one template sample of each subject, resulting in the single sample per person (SSPP) problem [175]. SSPP makes FR much more difficult because we have little information from the gallery set to predict the variations in the query face image [213].

Since the intra-class variations cannot be well estimated in the SSPP problem, the traditional discriminative subspace learning based FR methods can fail to work. In addition, since the number of samples per class is so small, the robustness of extracted features and the generalization ability of learned classifiers can be much reduced. To alleviate these difficulties of FR with SSPP, researchers have proposed to generate virtual samples of each subject, extract more discriminative features, and learn the facial variations from external data, etc. Generally speaking, the existing FR methods for SSPP can be categorized into three groups: virtual sample generation, generic learning and patch/block based methods.

Virtual sample generation aims to estimate the intra-class face variations by simulating extra samples for each subject. Virtual samples can be generated by perturbation-based approaches [122], geometric transform and photometric changes [165], SVD decomposition [54] and 3D methods [185], etc. With the virtual sam-

ples, intra-class scatter can be calculated to make Fisher linear discriminant analysis feasible in the scenario of SSPP [122][165][54]. Although virtual samples are helpful to FR with SSPP, they are highly correlated with the original face images and cannot be considered as independent samples for feature extraction. Therefore, there may exist much redundancy in the learned discriminative feature subspace [122][113].

Considering the similarity of face images across subjects, a generic training set can be used to compensate for the shortage of samples in FR. On one hand, the face variation information in the generic training set can be used to learn a projection matrix to extract discriminative features [97][189][93][128]. In [97] and [128], discriminative pose-invariant and expression-invariant projection matrices are learned by using a collected generic training set for pose-invariant and expression-invariant FR tasks, respectively. On the other hand, the abundant intra-class variations in the generic training set are very useful to more accurately represent a query face with unknown variations [43][213][81]. The sparse representation based classification (SRC) [201] represents a query face as a sparse linear combination of training samples from all classes. SRC shows interesting FR results; however, its performance will deteriorate significantly when the number of training samples of each class is very small because in such cases the variation space of each subject cannot be well spanned. The extended SRC (ESRC) [43] constructs an intra-class variation dictionary to represent the changes between the gallery and query images. In the case of SSPP, Yang et al. [213] learned a sparse variation dictionary by taking the relationship between the gallery set and the external generic set into account. The so-called sparse variation dictionary learning (SVDL) scheme shows state-of-the-art perfor-

mance in FR with SSPP. However, SVDL ignores the distinctiveness of different parts of human faces.

Patch/block based methods [27][113][102][231] [101] partition each face image into several patches/blocks, and then perform feature extraction and classification on them. First, patches can be viewed as independent samples for feature extraction [27][113]. In [27], the patches of each subject are considered as the samples of this class and then the within-class scatter matrix can be computed. In [113], the patches of each subject are considered to form a manifold and a projection matrix is learned by maximizing the manifold margin. Second, a weak classifier can be obtained from each patch, and then the classifiers on all patches can be combined to output the final decision (i.e., a strong classifier) [102][231]. In [102], the nearest neighbor classifier (NNC) is used for classification on each patch, and a kernel plurality method is proposed to combine the decisions on all patches. In [231], the collaborative representation based classifier (CRC) [226] is applied to each patch, and the majority voting is used for decision combination. Although the patch based methods in [102] and [231] significantly improve the FR performance compared with the original NNC and CRC classifiers, respectively, they do not solve the problem of lacking facial variations in the gallery set.

In this chapter, we propose a local generic representation (LGR) based scheme for FR with SSPP, whose framework is illustrated in Fig. 3.1. The training samples in the gallery set are used to build a gallery dictionary. To introduce the face intra-class variation information that is lacked in the gallery set, a generic training set, which contains a reference subset and several variation subsets, is collected. A generic variation dictionary is then constructed as the difference between the refer-

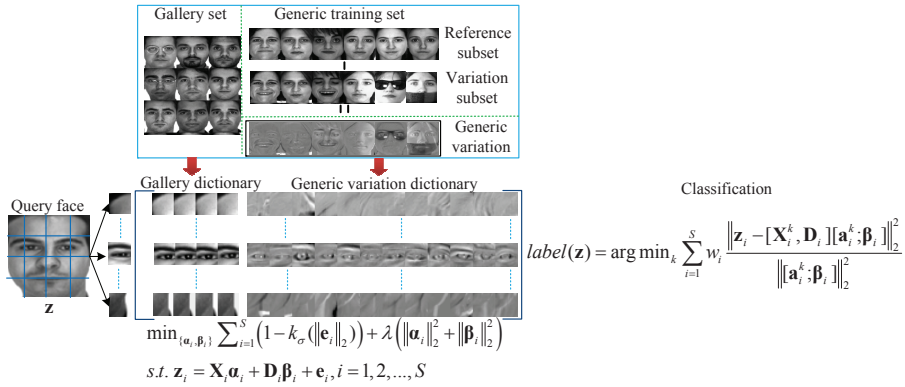


Figure 3.1 Framework of local generic representation based classification.

ence subset and the variation subsets. Considering the different importance of different facial parts in FR, we adopt a local representation approach, i.e., each patch of the query sample is represented by the patch gallery dictionary and patch variation dictionary at the corresponding location. LGR aims to minimize the total representation residual of all patches. Since the residuals are non-Gaussian distributed, we use correntropy to measure the loss in minimization. The half-quadratic optimization technique is used to solve the optimization problem. Finally, the classification is performed based on the overall representation residual of the query sample by each class. The experimental results on benchmark face databases, including Extended Yale B [59], CMU Multi-PIE [64], AR [123] and LFW [82], show that LGR outperforms many state-of-the art methods for FR with SSPP.

The rest of the chapter is organized as follows. Section 4.2 introduces the model of local generic representation. Section 4.3 discusses the model optimization and classification scheme. Section 4.4 conducts experiments and conclusions are made in Section 4.5.

3.2 Local generic representation

3.2.1 Generic representation

In FR with SSPP, we have a gallery set $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_k, ..., \boldsymbol{x}_K] \in \mathbb{R}^{d \times K}$, where $\boldsymbol{x}_k \in \mathbb{R}^d$ is the only single gallery sample of class k, $k = 1, 2, ..., K$. Given a query sample $\boldsymbol{z} \in \mathbb{R}^d$, representation based classifiers such as SRC [201] represent it over the gallery set $\boldsymbol{X}$ as:

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\alpha} + \boldsymbol{e} \tag{3.1}$$

If the gallery set has many training samples for each subject, most of the facial variations in the query sample can be synthesized by the multiple samples from the same class, and consequently correct classification can be made via comparing the representation residual of each class. For FR with SSPP, unfortunately, there is only one training sample per subject, and the variations (e.g., illumination, pose, expression, etc.) in $\boldsymbol{z}$ cannot be well represented by the single same-class sample in $\boldsymbol{X}$. Thus, the representation residual of $\boldsymbol{z}$ can be big, and $\boldsymbol{z}$ can be wrongly represented by samples from other classes, leading to misclassification of $\boldsymbol{z}$. Fig. 3.2(a) shows an example. The query image has some illumination change compared with the single gallery sample of its class. We use the SRC model to solve the representation in Eq. (3.1), i.e., $\min_{\alpha} \|\boldsymbol{z} - \boldsymbol{X}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$. One can see from Fig. 3.2(a) that the synthesized image $\boldsymbol{X}\boldsymbol{\alpha}$ does not overcome the problem of illumination change, and the illumination change is put forward into the representation residual $\boldsymbol{e}$. Such a representation will cause trouble in the classification stage.

Considering that the intra-class facial variations caused by illumination, pose, and expression changes and disguise can be shared across subjects, an external

(a) sparse representation (b) generic representation

Figure 3.2 Sparse representation versus generic representation.

generic training set which consists of enough face images with various types of variations can be adopted to construct an intra-class variation dictionary [43][213]. Suppose that we have collected a generic training set $\boldsymbol{G} = [\boldsymbol{G}^r, \boldsymbol{G}^v]$, where $\boldsymbol{G}^r$ and $\boldsymbol{G}^v$ are the reference subset and variation subset, respectively. The reference subset $\boldsymbol{G}^r \in \mathbb{R}^{d\times n}$ is composed of neutral face images or the mean faces of each subject. The variation subset $\boldsymbol{G}^v$ involves M possible facial variations: $\boldsymbol{G}^v = [\boldsymbol{G}^v_1, ..., \boldsymbol{G}^v_m, ..., \boldsymbol{G}^v_M]$, where $\boldsymbol{G}^v_m$ is the subset of the m^{th} variation, $m = 1, 2, ..., M$. In [213], a sparse variation dictionary is learned from $\boldsymbol{G}$. In our work, we simply construct an intra-class variation dictionary, denoted by $\boldsymbol{D}$, by using the difference between $\boldsymbol{G}^r$ and $\boldsymbol{G}^v$:

$$\boldsymbol{D} = [\boldsymbol{G}^v_1 - \boldsymbol{G}^r, ..., \boldsymbol{G}^v_m - \boldsymbol{G}^r, ..., \boldsymbol{G}^v_M - \boldsymbol{G}^r] \in \mathbb{R}^{d\times nM} \tag{3.2}$$

We then propose to represent the query sample $\boldsymbol{z}$ over the gallery set $\boldsymbol{X}$ and the generic variation dictionary $\boldsymbol{D}$ simultaneously:

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\alpha} + \boldsymbol{D}\boldsymbol{\beta} + \boldsymbol{e} \tag{3.3}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the representation vectors of $\boldsymbol{z}$ over $\boldsymbol{X}$ and $\boldsymbol{D}$, respectively, and $\boldsymbol{e}$ is the representation residual. We call the representation in Eq. (3.3) generic representation, which uses a generic intra-class variation dictionary $\boldsymbol{D}$ to account for the variations in the query sample. Fig. 3.2(b) shows the generic representation

of the query sample in Fig. 3.2(a). We use the following model to solve Eq. (3.3): $\min_{\{\alpha,\beta\}} ||\boldsymbol{z} - \boldsymbol{X}\boldsymbol{\alpha} - \boldsymbol{D}\boldsymbol{\beta}||_2^2 + \lambda(||\boldsymbol{\alpha}||_1 + ||\boldsymbol{\beta}||_1)$. One can clearly see that the illumination change in the query sample is well encoded by the generic variation dictionary $\boldsymbol{D}$, and the residual $\boldsymbol{e}$ has much lower energy ($||\boldsymbol{e}||_2^2$=0.0049) than the residual in Fig. 3.2(a) ($||\boldsymbol{e}||_2^2$=0.0502).

3.2.2 Patch based local generic representation

Different parts (e.g., eye, mouth, nose, cheek) of human faces exhibit distinct structures, and they have different importance in identifying the identity of a face. Taking this fact into account, we propose to localize the representation model in Eq. (3.3) and present a patch based local generic representation scheme.

We partition the query sample $\boldsymbol{z}$ into S (overlapped) patches and denote these patches as $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_S\}$. Correspondingly, the gallery dictionary $\boldsymbol{X}$ and the generic variation dictionary $\boldsymbol{D}$ can be partitioned as $\left\{\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_S\right\}$ and $\left\{\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_S\right\}$, respectively. For each local patch $\boldsymbol{z}_i, i = 1, 2, .., S$, its associated local gallery dictionary and local variation dictionary are $\boldsymbol{X}_i$ and $\boldsymbol{D}_i$, respectively. To increase the representation power of local gallery dictionaries and better address the local deformation (e.g., misalignment) of a patch, we extract the neighborhood patches at location i from each gallery sample, and add them to $\boldsymbol{X}_i$. Such a sample expansion of local gallery dictionaries can improve much the stability and robustness of local representation [231]. In our implementation, the 8 closet neighboring patches to the underlying patch at location i are extracted. With $\boldsymbol{X}_i$ and $\boldsymbol{D}_i$, we can represent each

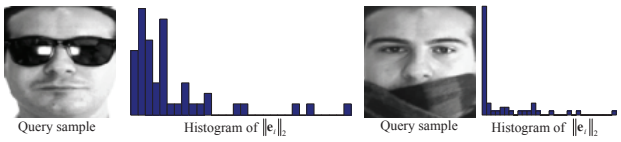


Figure 3.3 The histogram of $\|\boldsymbol{e}_i\|_2, i = 1, 2, ..., S$, for two query samples.

local patch $\boldsymbol{z}_i$ as:

$$\boldsymbol{z}_i = \boldsymbol{X}_i\boldsymbol{\alpha}_i + \boldsymbol{D}_i\boldsymbol{\beta}_i + \boldsymbol{e}_i, i = 1, 2, ..., S \tag{3.4}$$

where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ are the representation vectors of $\boldsymbol{z}_i$ over $\boldsymbol{X}_i$ and $\boldsymbol{D}_i$, respectively, and $\boldsymbol{e}_i$ is the representation residual.

Clearly, in order to find meaningful solutions of vectors $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$, appropriate loss function should be defined on the representation residual $\boldsymbol{e}_i$ and appropriate regularization can be imposed on $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$. Denote by $l(\|\boldsymbol{e}_i\|_2)$ the loss function defined on the l_2-norm of $\boldsymbol{e}_i$ and denote by $R(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ some regularizer imposed on the representation coefficients. We consider the following optimization problem to solve $\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\}$:

$$\begin{aligned} &\min_{\{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i\}} \textstyle\sum_{i=1}^{S} l(\|\boldsymbol{e}_i\|_2) + \lambda R(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) \\ &s.t.\ \boldsymbol{z}_i = \boldsymbol{X}_i\boldsymbol{a}_i + \boldsymbol{D}_i\boldsymbol{\beta}_i + \boldsymbol{e}_i,\ i = 1, 2, ..., S \end{aligned} \tag{3.5}$$

The problem now turns to how to define the loss function $l(\|\boldsymbol{e}_i\|_2)$ and regularizer $R(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$. Let $e_i = \|\boldsymbol{e}_i\|_2$. Due to the special structure of human face, the different patches will have very different representation residuals e_i. We solve $\{\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i\} = \min_{\{\boldsymbol{\alpha}_i,\boldsymbol{\beta}_i\}} \|\boldsymbol{z}_i - \boldsymbol{X}\boldsymbol{\alpha}_i + \boldsymbol{D}_i\boldsymbol{\beta}_i\|_2^2 + \lambda(\|\boldsymbol{\alpha}_i\|_2^2 + \|\boldsymbol{\beta}_i\|_2^2)$ and then calculate $e_i = \left\|\boldsymbol{z}_i - \boldsymbol{X}\hat{\boldsymbol{\alpha}}_i + \boldsymbol{D}_i\hat{\boldsymbol{\beta}}_i\right\|_2$. Fig. 3.3 illustrates the distribution for e_i for two query face images. One can see that the distribution of e_i is highly non-Gaussian. The widely used l_2-norm loss function relies highly on the Gaussianity assumption of the

data [112] and hence it is not suitable to measure such non-Gaussian distributed residual. In [110], the concept of correntropy is proposed to measure the loss of non-Gaussian data. A correntropy induced metric (CIM) for residual e_i is defined as [110]:

$$\mathrm{CIM}(e_i) = (k_\sigma(0) - k_\sigma(e_i))^{1/2} \tag{3.6}$$

where $k_\sigma(\cdot)$ is a kernel function. The Gaussian kernel function $k_\sigma(x) = \exp(-x^2/2\sigma^2)$ is widely used with good performance [110] [112]. The robustness of CIM to non-Gaussian residual/noise has been verified in signal processing [134], feature selection [72], and FR [74]. Hence, we adopt correntropy to model the representation residual of different patches.

For the regularizer $R(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$, we define it as the l_2-norm of $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$. It has been shown that the l_2-norm regularization on representation coefficients can lead to similar classification performance to l_1-norm regularization but with much less computational cost [226]. Finally, the proposed local generic representation (LGR) model becomes:

$$\begin{aligned} &\min_{\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\}} \sum_{i=1}^{S} (1 - k_\sigma(\|\boldsymbol{e}_i\|_2)) + \lambda \left(\|\boldsymbol{\alpha}_i\|_2^2 + \|\boldsymbol{\beta}_i\|_2^2 \right) \\ &s.t. \boldsymbol{z}_i = \boldsymbol{X}_i \boldsymbol{\alpha}_i + \boldsymbol{D}_i \boldsymbol{\beta}_i + \boldsymbol{e}_i, i = 1, 2, ..., S \end{aligned} \tag{3.7}$$

3.3 Optimization and classification

3.3.1 Half-quadratic optimization

The minimization problem in Eq. (3.7) can be solved by half-quadratic optimization [134]. If a function $\phi(x)$ satisfies the following conditions [134]: (a) $x \to \phi(x)$ is convex on $\mathbb{R}$; (b) $x \to \phi(\sqrt{x})$ is concave on $\mathbb{R}_+$; (c) $\phi(x) = \phi(-x), x \in \mathbb{R}$; (d)

$x \to \phi(x)$ is C^1 on $\mathbb{R}$; (e) $\phi''(0^+) > 0$; (f) $\lim_{x\to\infty}\phi(x)/\|x\|_2^2 = 0$, there exists a dual function φ such that

$$\phi(x) = \inf_{w\in\mathbb{R}} \left\{\frac{1}{2}wx^2 + \varphi(w)\right\} \tag{3.8}$$

where w is determined by the minimizer function $\delta(\cdot)$ with respect to $\phi(\cdot)$. $\delta(\cdot)$ admits an explicit form under certain restrictive assumptions [134]:

$$w = \begin{cases} \delta(t) = \phi''(0^+), \text{ if } t = 0 \\ \phi''(t)/t, \qquad\quad \text{if } t \neq 0 \end{cases} \tag{3.9}$$

Obviously, $\phi_\sigma(x) = 1 - k_\sigma(x) = 1 - \exp(-x^2/2\sigma^2)$ satisfies all the conditions from (a) to (f). Then the problem in Eq. (3.7) can be equivalently written as the following augmented minimization problem:

$$\min_{\boldsymbol{A},\boldsymbol{w}} \sum_{i=1}^{S} \left(\frac{1}{2}w_i \|\boldsymbol{z}_i - \boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{D}_i\boldsymbol{\beta}_i\|_2^2 + \varphi(w_i)\right) + \lambda\|\boldsymbol{A}\|_2^2 \tag{3.10}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_S]$ with $\boldsymbol{a}_i = [\boldsymbol{\alpha}_i; \boldsymbol{\beta}_i]$, and $\boldsymbol{w} = [w_1, w_2, ..., w_S]$.

According to the half-quadratic analysis [134], Eq. (3.10) can be easily minimized by updating $\boldsymbol{A}$ and $\boldsymbol{w}$ alternatively, and there is no need to have an explicit form of the dual function $\varphi(w_i)$. When $\boldsymbol{w}$ is fixed, $\boldsymbol{A}$ can be solved by

$$\hat{\boldsymbol{A}} = \arg\min_{\boldsymbol{A}} \sum_{i=1}^{S} \left(w_i \|\boldsymbol{z}_i - \boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{D}_i\boldsymbol{\beta}_i\|_2^2\right) + \lambda\|\boldsymbol{A}\|_F^2 \tag{3.11}$$

Clearly, the above minimization is a least square regression problem, and we have the closed-form solution of each $\{\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i\}$:

$$[\hat{\boldsymbol{\alpha}}_i; \hat{\boldsymbol{\beta}}_i] = w_i(w_i[\boldsymbol{X}_i, \boldsymbol{D}_i]^T[\boldsymbol{X}_i, \boldsymbol{D}_i] + \lambda\boldsymbol{I})^{-1}[\boldsymbol{X}_i, \boldsymbol{D}_i]^T\boldsymbol{z}_i \tag{3.12}$$

When $\boldsymbol{A}$ is fixed, the weights $\boldsymbol{w}$ can be updated as

$$\hat{w}_i = \frac{1}{\sigma^2}\exp(-\|\boldsymbol{z}_i - \boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{D}_i\beta_i\|_2^2/2\sigma^2) \tag{3.13}$$

The weight w_i corresponds to the i^{th} patch, and it is used to control the portion of $\|\boldsymbol{e}_i\|_2$ in the whole energy of Eq. (3.10). If the representation residual of a patch is big (e.g., caused by sunglasses, scarf and/or other large variations), the corresponding weight w_i will become small, and consequently the effect of this patch in the overall representation will be suppressed.

3.3.2 LGR based classification

After the optimal solutions of $\boldsymbol{A}$ and $\boldsymbol{w}$ are resolved by the half-quadratic optimization in Section 3.3.1, an LGR based classification scheme can be proposed to determine the class label of query face $\boldsymbol{z}$. Let $\boldsymbol{X}_i = \left[\boldsymbol{X}_i^1, ..., \boldsymbol{X}_i^k, ..., \boldsymbol{X}_i^K\right]$, where $\boldsymbol{X}_i^k$ is sub-gallery dictionary associated with class k. Accordingly, the representation vector $\boldsymbol{\alpha}_i$ can be written as $\boldsymbol{\alpha}_i = \left[\boldsymbol{\alpha}_i^1; ...; \boldsymbol{\alpha}_i^k; ...; \alpha_i^K\right]$, where $\boldsymbol{\alpha}_i^k$ is the coefficients vector associated with class k. By using the class-specific sub-gallery dictionary $\boldsymbol{X}_i^k$ and the generic variation dictionary $\boldsymbol{D}_i$, we can calculate the representation residual of each patch $\boldsymbol{z}_i$ by each class k. Then the sum of the weighted residual (by w_i) over all patches can be calculated. Our classification principle is to check which class can lead to the minimal residual over all patches. Specifically, the classification rule of query face $\boldsymbol{z}$ is as follows:

$$label(\boldsymbol{z}) = \arg\min_k \sum_{i=1}^{S} w_i \left\|\boldsymbol{z}_i - [\boldsymbol{X}_i^k, \boldsymbol{D}_i][\boldsymbol{a}_i^k;\boldsymbol{\beta}_i]\right\|_2^2 / \left\|[\boldsymbol{a}_i^k;\boldsymbol{\beta}_i]\right\|_2^2 \tag{3.14}$$

Note that in Eq. (3.14), we also use the l_2-norm of $[\boldsymbol{a}_i^k;\boldsymbol{\beta}_i]$ to adjust the residual of patch i by class k. $1/\left\|[\boldsymbol{a}_i^k;\boldsymbol{\beta}_i]\right\|_2^2$ can be considered as a "class weight". If class k has a larger $\left\|[\boldsymbol{a}_i^k;\boldsymbol{\beta}_i]\right\|_2^2$, it means that the query patch is more similar to the gallery patch of class k, and thus a smaller weight should be assigned to weaken the repre-

Table 3.1 The algorithm of local generic representation (LGR) based classification.

Input: The query sample $\boldsymbol{z}$, gallery set $\boldsymbol{X}$, reference subset $\boldsymbol{G}^r$, variation subset $\boldsymbol{G}^v$ and regularization parameter λ.

Output: The class label of $\boldsymbol{z}$

1: Initialize $\boldsymbol{w} = [1, 1, ..., 1]$;

2: Caculate $\boldsymbol{D} = [\boldsymbol{G}_1^v - \boldsymbol{G}^r, \boldsymbol{G}_2^v - \boldsymbol{G}^r, ..., \boldsymbol{G}_m^v - \boldsymbol{G}^r]$.

3: Partition $\boldsymbol{z}$, $\boldsymbol{X}$ and $\boldsymbol{D}$ into patches.

4: While convergence

5: Update $\boldsymbol{A}$ by Eq. (3.11);

6: Update $\boldsymbol{w}$ by Eq. (3.13);

7: End

8: Output the class label of sample $\boldsymbol{z}$ by Eq.(3.14).

sentation residual by this class. The query sample $\boldsymbol{z}$ is classified to the class which has the minimal weighted representation residual over all patches. The algorithm of LGR based classification is summarized in Table 3.1.

3.3.3 Convergence and complexity

According to half-quadratic optimization [134], the objective function in Eq. (3.10) is non-increasing under the update rules in Eq.(3.11) and Eq. (3.13). Therefore, our algorithm is guaranteed to converge based on the theory of half-quadratic optimization [134]. In Fig.3.4, the convergence curve of LGR on the AR database [123] is shown (please refer to section 3.4.4 for the details of experiment setting). We can see that the LGR algorithm converges after 5 iterations.

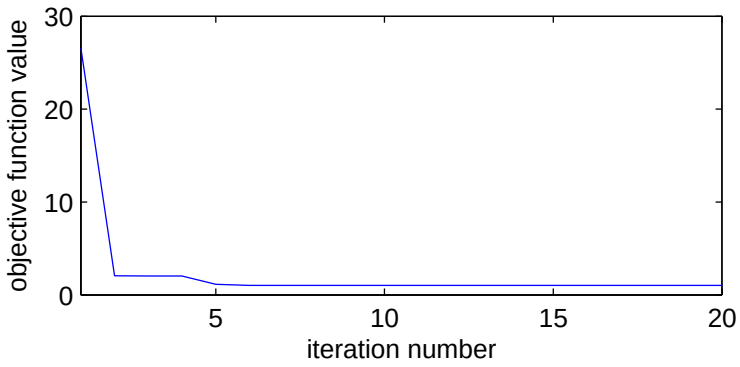


Figure 3.4 The convergence curve of LGR on the AR database.

The main computational cost of LGR is spent on solving the least square regression problem in Eq. (3.11), whose time complexity is $O(S(n_d^3 + n_d^2 d_p))$, where S is the number of patches, n_d is the total number of patches in $[\boldsymbol{X}_i, \boldsymbol{D}_i]$ and d_p is the feature dimension of patches. Denote by T the total number of iteration in our algorithm, the time complexity of LGR is $O(TS(n_d^3 + n_d^2 d_p))$.

3.4 Experimental analysis

We test the performance of LGR on four benchmark face databases, including three face databases in controlled environment, i.e., Extended Yale B [59], large-scale CMU Multi-PIE [64], and AR [123], and one face database in uncontrolled environment, i.e., Labeled Faces in the Wild (LFW) database [82]. Extended Yale B database contains illumination variations; AR database contains illumination and expression variations and disguises; Multi-PIE database contains pose, illumination and expression variations; LFW reflects the variations in real-world applications. We compare the proposed LGR method with the following eleven methods:

- Baseline methods: nearest neighbor classifier (NNC) [38], support vector machines (SVM) [36], sparse representation based classifiers (SRC) [201] and collaborative representation based classifiers (CRC) [226];
- Generic learning methods: adaptive generic learning (AGL) [173], extended SRC (ESRC) [43] and sparse variation dictionary learning (SVDL) [213];
- Patch/block based methods: Block linear discriminative analysis (BlockLDA) [27], patch based NN (PNN) [102], patch based CRC (PCRC) [231], and discriminative multi-manifold analysis (DMMA) [113].

Note that the generic learning method SVDL learns a sparse variation dictionary from the generic training set. The proposed LGR also belongs to the generic learning methods; however, we use the raw face difference images as the dictionary rather than learning a dictionary with some objective function. Among the competing methods, we implement NN and DMMA; the code of SVM is from [24]; and the codes of all the other methods are obtained from the original authors.

3.4.1 Parameter setting

In all the experiments, the face images are resized to 80×80 (using the Matlab function "resize.m"). For patch/block based methods including BlockLDA, PNN, PCRC, DMMA, and the proposed LGR, the patch size is fixed as 20×20 and the overlap between neighboring patches is 10 pixels. That is, the query sample is partitioned into S =49 patches.

Apart from the setting of patch size and patch number, there are only two parameters to set in the proposed LGR. The first is the regularization parameter λ in

Eq. (6). We fix it as λ= 0.001 in all our experiments. Another is the scale parameter σ of the kernel function $k_\sigma(x)$. Based on our experimental experience, if the representation residual is big, a large value of σ could be set to make the representation more robust. Therefore, we adaptively set σ as the average representation residual after solving the coefficients $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ in the first iteration of our algorithm; that is, $\sigma = \sqrt{\frac{1}{2S}\sum_{i=1}^{S} \|\boldsymbol{z}_i - \boldsymbol{X}_i\boldsymbol{\alpha}_i - \boldsymbol{D}_i\boldsymbol{\beta}_i\|_2^2}$.

For the competing algorithms, we tune their parameters for the best results. In particular, for SVDL we follow the parameter setting in [213]. The three parameters λ_1, λ_2, λ_3 are set as 0.001, 0.01, 0.0001, respectively, and the number of dictionary atoms is set as 400 in the initialization. For SRC, CRC and PCRC, the optimal regularization parameter λ is chosen from $\{0.0005, 0.001, 0.005, 0.01\}$. As BlockLDA and AGL are sensitive to the feature dimension, the best result of different feature dimensions is reported.

3.4.2 Extended Yale B database

The Extended Yale B face database [59] contains 38 human subjects and 2,414 face images with 64 illumination conditions. The frontal faces with light source directions at 0 degree azimuth (A+000) and at 0 degree elevation (E+00) are used as the gallery set, and the face images under other illumination conditions are used as the query set. We use the face images of the first 30 subjects to form the gallery and query sets, and use the face images of the other 8 subjects as the generic set.

Table 3.2 lists the recognition rates by different methods. By combining the decisions of different patches, the PCRC method achieves much higher recognition rate than the baseline methods. The generic learning based method SVDL achieves

Figure 3.5 Face images of Extended YaleB database.

the second highest recognition rate by learning a dictionary that consists of different illumination variations. By exploiting the advantages of both patch based local representation and generic variation information, the proposed LGR method achieves the highest recognition accuracy.

Table 3.2 Recognition rate (%) on Extended Yale B database.

Method	NNC[38]	SVM[36]	SRC[201]	CRC[226]	BlockLDA[27]	AGL[173]
Accuracy	46.5	41.4	49.2	51.2	49.2	59.5
Method	DMMA[113]	PNN[102]	PCRC[231]	ESRC[43]	SVDL[213]	LGR
Accuracy	61.7	67.5	77.8	67.9	85.0	**86.6**

3.4.3 CMU Multi-PIE database

The Multi-PIE database [64] contains a total of more than 750,000 images from 337 individuals, captured under 15 viewpoints and 19 illumination conditions in four recording sessions. The face images of the first 100 subjects in session 1 are used for the gallery set and the other 149 subjects are used as generic set. Following the experiment setting in [213], in the generic training set, the frontal images with illumination 7 and neutral expression are used as the reference subset and the face images with different variations in Session 1 are used as the variation subset.

Illumination variations

In this experiment, we test the performance of LGR under different illuminations. The frontal face images with neutral expression from session 2, session 3 and session 4 are used as the query set, respectively. The recognition rates on Multi-PIE with illumination variations are listed in Table 3.3. LGR shows superior performance to all the other competing methods. Compared with SVDL, which achieves the second highest accuracy, the recognition rate is improved by 2.7%, 3.0% and 4.0% on session 2, session 3 and session 4, respectively. Compared with PCRC, the recognition rate is improved by about 15%. The performance of SRC and CRC is very poor because with only one gallery face image per person, the query image cannot be well represented.

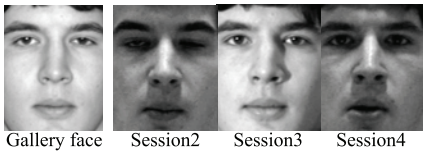


Figure 3.6 Images of Multi-PIE database with Illumination variations in different sessions.

Expression and illumination variations

We then test the robustness of the proposed LGR method to face images with both expression and illumination variations. The query set includes the frontal face images with smile expression in session 1 (Smile-S1), smile expression in session 3 (Smile-S3) and surprise expression (Surprise-S2). Table 3.4 presents the recognition results in this experiment. Clearly, LGR outperforms all the other methods.

Table 3.3 Recognition accuracy (%) on Multi-PIE with illumination variations.

Method	Session 2	Session 3	Session 4
NNC[38]	44.3	40	43.8
SVM[36]	43.6	40.5	40.1
SRC[201]	51.9	46.5	50.6
CRC[226]	52.8	47.4	50.5
BlockLDA[27]	68.2	60.4	65.1
AGL[173]	84.5	79.6	78.5
DMMA[113]	64.1	56.6	60.1
PNN[102]	65.1	55.6	60.8
PCRC[231]	83.7	72.7	77.7
ESRC[43]	92.6	84.6	87.6
SVDL[213]	94.2	87.5	90.4
LGR	**96.9**	**90.5**	**94.4**

SVDL still works the second best, but it lags behind LGR by 1.8%, 5.6% and 21.7% for Smile-S1, Smile-S3 and Surprise-S2, respectively.

Pose, expression and illumination variations

In this experiment, there are pose, expression and illumination variations in the query set simultaneously. We select the face images with pose 05_0 in Session 2 (P1), pose 04_1 in Session 3 (P2), and pose 04_1 and smile expression in Session 3 (P3) as the query set. Some face images from the gallery and query set are illustrated in Fig. 3.7.

Table 3.4 Recognition accuracy (%) on Multi-PIE with expression and illumination variations.

Method	Smile-S1	Smile-S3	Surprise-S2
NNC[38]	46.8	29.1	18.3
SVM[36]	46.8	29.1	18.3
SRC[201]	50.1	28.1	21.1
CRC[226]	50	29.7	22.4
BlockLDA[27]	49.5	30	26.2
AGL[173]	85.2	39.5	31.5
DMMA[113]	58.5	33.4	23
PNN[102]	53.1	31.1	31.4
PCRC[231]	74.9	44.1	44.9
ESRC[43]	82	50.8	49.9
SVDL[213]	88.9	59.6	52.8
LGR	**90.7**	**65.2**	**74.5**

Table 3.5 lists the recognition rate of all methods. LGR achieves the highest accuracy on all the three query sets. Because of the large variations caused by pose, expression and illumination variations, the FR rates in this experiment are relatively lower than the experimental results in Table 3.3 and Table 3.4. The patch based methods such as PCRC do not work well because they are sensitive to pose variation. The generic learning methods, including AGL, ESRC, SVDL and the proposed LGR, outperform the other methods since they can exploit the variation information from the external generic training set. LGR consistently exhibits better

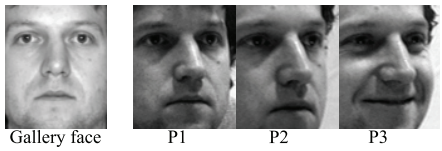


Figure 3.7 Images of Multi-PIE database with pose, expression and illumination variations.

results than SVDL, which still works the second best.

3.4.4 AR face database

The AR face database [123] contains about 4,000 color face images of 126 people, which consists of the frontal faces with different facial expressions, illuminations and disguises. There are two sessions and each session has 13 face images per subject. Following the SSPP experiment setting in [43], a subset with face images of 50 males and 50 females is selected. The first 80 subjects from sessions 1 are used for the gallery and query set while the other 20 subjects are used as the generic training set. We also use the face images from session 2 as the query set to test the FR performance. There are different variations, including illumination, expression, and disguise (scarf and sunglass) in this experiment.

The experimental results on session 1 and session 2 are shown in Table 3.6 and Table 3.7, respectively. LGR exhibits significantly better performance than all the other methods on both sessions. In particular, on session 2 LGR outperforms SVDL by 16.4%, 10.8%, 32.5% and 34.7% under different variations. Note that in this experiment the performance of patch based methods such as PCRC is very competitive. This is because the disguises (i.e., scarf and sunglass) can be well dealt with by patch/block based methods. Therefore, PCRC can achieve higher

Table 3.5 Recognition accuracy (%) on Multi-PIE with pose, expression and illumination variations.

Method	P1	P2	P3
NNC[38]	25.7	8.8	11.9
SVM[36]	25.7	8.8	11.9
SRC[201]	23.9	6.1	10.1
CRC[226]	24.9	5.4	9.0
BlockLDA[27]	29.5	13.2	15.8
AGL[173]	66.4	25.5	24.0
DMMA[113]	28.2	5.5	12.1
PNN[102]	35.3	11.8	13.5
PCRC[231]	37.3	8.0	10.2
ESRC[43]	63.8	31.9	27.0
SVDL[213]	76.0	37.9	33.5
LGR	**79.1**	**39.5**	**36.3**

recognition rate than the global representation based SVDL though it does not learn any variation information from a generic dataset. The proposed LGR utilizes both local presentation and generic information, leading to very promising performance for the task of FR with SSPP.

3.4.5 LFW database

The LFW database [82] contains images of 5,749 different individuals in unconstrained environment. LFW-a is a version of LFW after alignment using commer-

Table 3.6 Recognition accuracy (%) on AR face database (session1).

Method	illumination	expression	disguise	illumination+disguise
NNC[38]	70	79.2	39.4	23.5
SVM[36]	55.8	90.4	43.1	29.4
SRC[201]	80.8	85.4	55.6	25.3
CRC[226]	80.5	80.4	58.1	23.8
BlockLDA[27]	75.3	81.4	65.4	53.5
AGL[173]	93.3	77.9	70.0	53.8
DMMA[113]	92.1	81.4	46.9	30.9
PNN[102]	84.6	86.7	90.0	72.5
PCRC[231]	95.0	86.7	95.6	81.3
ESRC[43]	99.6	85.0	83.1	68.6
SVDL[213]	98.3	86.3	86.3	79.4
LGR	**100**	**97.9**	**98.8**	**96.3**

cial face alignment software [197]. Following the experiment setting in [231] and [213], a subset of 158 subjects with more than 10 images per person is collected. Each face image is cropped to 120×120 and then resized to 80×80. Fig. 3.8 shows some face images in the LFW-a dataset. One can see that although face alignment has been conducted, the variations in this database is still very large compared with the face databases in the controlled environment. Face images of the first 50 subjects are selected to form the gallery and query sets, while the face images of the remaining subjects are used to build the generic training set. Since there are no frontal neutral face images in this database, the mean face of each person is used to

Table 3.7 Recognition accuracy (%) on AR face database (session2).

Method	illumination	expression	disguise	illumination+disguise
NNC[38]	41.7	58.8	26.3	12.8
SVM[36]	40.0	58.8	26.9	14.4
SRC[201]	55.8	68.8	29.4	12.8
CRC[226]	55.8	69.6	35.0	13.5
BlockLDA[27]	54.7	61.2	31.9	21.0
AGL[173]	70.8	55.8	40.6	30.7
DMMA[113]	77.9	61.7	28.1	21.9
PNN[102]	77.5	73.8	71.9	52.8
PCRC[231]	88.8	71.7	81.8	63.1
ESRC[43]	87.9	70.4	59.4	45.0
SVDL[213]	87.1	74.2	61.3	54.1
LGR	**97.5**	**85.0**	**93.8**	**88.8**

form the reference subset in the generic set.

The face recognition rates of different methods are listed in Table 3.8. Because of the challenging face variations in uncontrolled environment, no method achieves very high accuracy in this experiment. Nonetheless, LGR still works the best among all competing methods. The patch based method PCRC works better than the global representation based CRC, which is similar to what we observed in the experiments of previous sections. SVDL again achieves the second highest recognition rate, demonstrating that the face variation information learned from other subjects is indeed helpful to improve the robustness of FR with SSPP, no matter in controlled

Figure 3.8 Images of LFW database.

or uncontrolled environment.

Table 3.8 Recognition accuracy (%) on LFW database.

Method	NNC[38]	SVM[36]	SRC[201]	CRC[226]	BlockLDA[27]	AGL[173]
Accuracy	12.2	11.6	20.4	19.8	16.4	19.2
Method	DMMA[113]	PNN[102]	PCRC[231]	ESRC[43]	SVDL[213]	LGR
Accuracy	17.8	17.6	24.2	27.3	28.6	**30.4**

3.5 Conclusions and discussions

We proposed a local generic representation (LGR) based approach for the challenging task of face recognition with single sample per person (SSPP). LGR utilizes the advantages of both patch based local representation and generic learning. A generic intra-class variation dictionary was constructed from a generic dataset, and it can well compensate for the face variations lacked in the SSPP gallery set. A patch gallery dictionary was built by using the gallery samples, which can more accurately represent the different parts of face images. Considering that the distribution of representation residual of different patches is highly non-Gaussian, a correntropy

based metric was adopted to measure the loss of each patch so that the importance of different patches in face recognition can be more robustly evaluated. As a result, LGR can adaptively suppress the role of patches with large variations. The extensive experimental results on four benchmark face databases showed that LGR always achieves higher face recognition rate than the state-of-the-art SSPP methods used in competition.

In this chapter, as generic training set introduces more across-subject face variations, the recognition performance of LGR is much better than PCRC. However, as LGR has to solve a half-quadratic optimization problem for a query face image, PCRC is much faster than LGR. Hence, in real-world applications, to solve face recognition with single sample per person, we can choose PCRC and LGR according to different demands.

Chapter 4

Regularized Self-Representation for Classification

In Chapter 2 and Chapter 3, we aim to solve small sample size problem in classification tasks. Whereas, with the development of sensors and digital devices, the size of available data is rapidly increasing. In some cases, there are a large amount of samples in the training dataset. For representation based classifiers, the solution will become less stable if the sample size is big, and the computation complexity and storage burden are quite high. Besides, the existing representation based models all belong to sample-level representation, i.e., a query sample is represented as a linear combination of training samples. Similarly, a query feature can also be represented by its related features. In this chapter, we aim to develop effective and efficient representation based classifier for big sample size classification task from the viewpoint of feature-level representation.

4.1 Introduction

Nearest neighbor classifier (NNC) has been widely used in machine learning and pattern recognition tasks such as face recognition [180], handwritten digit recognition [106], and image classification [15], etc. NNC measures the distance/similarity between the query sample and each of the training samples independently, and assigns the label of the nearest sample to the query sample. If the training samples are distributed densely enough, the classification error of NNC is bounded by twice the classification error of Bayesian classifier [38]. NNC does not need the prior knowledge of sample distribution and it is parameter-free. However, NNC ignores the relationship between training samples [186], and often fails for high-dimensional pattern recognition tasks because of the curse of dimensionality [150]. Besides, all training samples should be stored in NNC and it becomes time-consuming in large scale problems [42].

To reduce the computation burden of NNC and dilute the curse of dimensionality, nearest subspace classifier (NSC) is proposed. NSC measures the distance from the query sample to the subspace of each class and then classifies the query sample to its nearest subspace. The subspaces are often used to describe the appearance of objects under different lighting [9], viewpoint [182][178], articulation [16][179], and identity [13]. Each class can be modeled as a linear subspace [31], affine hull (AH) [186] or convex hull (CH) [186], hyperdisk [22] or variable smooth manifold [111]. When one class is considered as a linear subspace, NSC actually represents a query sample by a linear combination of the samples in that class. In such a case, a set of projection matrices can be calculated offline, and thus NSC avoids the one-to-one searching process in NNC, reducing largely the time cost.

Some approximate nearest subspace algorithms have also been proposed to further accelerate the searching process [8]. Whereas, NSC only considers the information of one class when calculating the distance from the query sample to this class, and it ignores the information of other classes.

As a significant extension to NSC, the sparse representation based classifier (SRC) [201] exploits the information from all classes of training samples when representing the given query sample, and it has shown promising classification performance [201]. Specifically, SRC represents the query sample as a linear combination of all training samples with l_1-norm sparsity constraint imposed on the representation coefficients, and then it classifies the query sample to the class with the minimal representation error [201]. In spite of the promising classification accuracy, SRC has to solve an l_1-norm minimization problem for each query sample, which is very costly. It has been shown in [226] that the collaborative representation mechanism (i.e., using samples from all classes to collaboratively represent the query image) plays a more important role in the success of SRC. By using l_2-norm to regularize the representation coefficients, the so-called collaborative representation based classification (CRC) demonstrates similar classification rates to SRC [201]. CRC has a closed-form solution to representing the query sample, and therefore has much lower computational cost than SRC.

Inspired by SRC and CRC, in [30] a collaborative representation optimized classifier (CROC) is proposed to pursue a balance between NSC and CRC. In [214], feature weights are introduced to the representation model to penalize pixels with large error so that the model is robust to outliers. A kernel sparse representation model is proposed by mapping features to a high dimensional reproducing kernel

Hilbert space [55]. In addition, dictionary learning methods have been proposed to learn discriminative dictionaries for representation based classifiers [90][215][115].

Most of the current representation based classifiers, including NSC, SRC and CRC, are sample oriented, and they represent a query sample as a combination of training samples. The time and memory complexity of such a "sample oriented" representation strategy, however, will increase rapidly with the number of training samples. For instance, in the training stage the time complexities of NSC and CRC are $O(Kn^3)$ and $O((Kn)^3)$, respectively, where K is the number of classes and n is the number of samples per class. Clearly, the complexity is exponential w.r.t. the training sample number. In the testing stage, the memory complexities of NSC and CRC are both $O(dKn)$, where d is the feature dimension. It is linear to the number of training sample and can be very costly for large scale pattern classification problems, where there are many classes and a lot of samples per class.

Different from those previous representation based classifiers, in this chapter we investigate the representation based classification problem from a "feature oriented" perspective. Instead of representing a sample as the linear combination of other samples, we propose to learn how each feature (i.e., each element) of a sample can be represented by the features of itself. Such a self-representation property of features generally holds for most high dimensional data, and has been applied in machine learning and computer vision fields [127]. For example, in [127] this property is used to select the representative features by feature clustering. Motivated by the self-representation property of sample features, we propose a novel self-representation induced classifier (SRIC), which learns a self-representation matrix for each class by its training data. To classify a query sample, we project it onto

the learned self-representation matrix and compute its feature self-representation residual. The query sample is then classified to the class which has minimal feature self-representation residual. Interestingly, it can be proved that SRIC is equivalent to NSC with l_2-norm regularization in terms of the final classification decision. Furthermore, it can be shown that SRIC is essentially the principle component analysis (PCA) with eigenvalue shrinkage.

SRIC learns the self-representation matrix individually for each class. In light of the principle of SRIC, we then present a discriminative SRIC (DSRIC) approach. Using all training data, for each class a discriminative self-representation matrix is trained to minimize the feature self-representation residual of this class while representing little the features of other classes. The classification of a query still depends on which class has the minimal feature self-representation residual. D-SRIC is intuitive and easy to understand. Our experimental results on UCI datasets, handwritten digit recognition, gender classification and face recognition show that DSRIC has comparable or superior recognition rate to state-of-the-art representation based classifiers such as SRC and CRC; however, our theoretical complexity analysis and experimental results will show that DSRIC is much more efficient and needs much less storage space than other representation based classifiers.

The rest of this chapter is organized as follows. Section 4.2 presents SRIC and analyzes its relationship with NSC and PCA. Section 4.3 presents the DSRIC method and analyzes its time and memory complexities in both training and testing stages. Section 4.4 conducts experiments on different pattern classification tasks, and Section 4.5 concludes.

4.2 Self-representation for classification

4.2.1 Nearest subspace classifier

Suppose that we have a set of training samples from K classes $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_k, ..., \boldsymbol{X}_K]$, where $\boldsymbol{X}_k = [\boldsymbol{x}_{1k}, ..., \boldsymbol{x}_{ik}, ..., \boldsymbol{x}_{nk}] \in \mathbb{R}^{d \times n}$, is the sample subset of class k and $\boldsymbol{x}_{ik}$ is the i^{th} sample of it, d is the feature dimension and n is the number of training samples in each class. Given a query sample $\boldsymbol{z}$, the nearest subspace classifier (NSC) represents it by the samples of class k as:

$$\boldsymbol{z} = \boldsymbol{X}_k \boldsymbol{a}_k + \boldsymbol{e}_k \tag{4.1}$$

where $\boldsymbol{a}_k$ is the representation vector and $\boldsymbol{e}_k$ is the representation residual vector.

To get an optimal representation of $\boldsymbol{z}$, NSC minimizes the representation residual by solving the following least square problem:

$$\hat{\boldsymbol{a}}_k = \arg\min_{\boldsymbol{a}_k} \|\boldsymbol{z} - \boldsymbol{X}_k \boldsymbol{a}_k\|_2^2 \tag{4.2}$$

The problem in Eq. (4.2) has a closed-from solution $\hat{\boldsymbol{a}}_k = (\boldsymbol{X}_k^T \boldsymbol{X}_k)^{-1} \boldsymbol{X}_k^T \boldsymbol{z}$ if $(\boldsymbol{X}_k^T \boldsymbol{X}_k)^{-1}$ is non-singular. In practice, an l_2-norm regularization can be imposed on $\boldsymbol{a}_k$ to make $(\boldsymbol{X}_k^T \boldsymbol{X}_k)^{-1}$ more stable, resulting in an l_2-norm regularized least regression problem:

$$\hat{\boldsymbol{a}}_k = \arg\min_{\boldsymbol{a}_k} \|\boldsymbol{z} - \boldsymbol{X}_k \boldsymbol{a}_k\|_2^2 + \lambda \|\boldsymbol{a}_k\|_2^2 \tag{4.3}$$

The analytical solution to Eq. (4.3) is $\hat{\boldsymbol{a}}_k = (\boldsymbol{X}_k^T \boldsymbol{X}_k + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}_k^T \boldsymbol{z}$, where $\boldsymbol{I}$ is an identity matrix. Then the representation residual can be computed as $r_k = \left\| \boldsymbol{z} - \boldsymbol{X}_k (\boldsymbol{X}_k^T \boldsymbol{X}_k + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}_i^T \boldsymbol{z} \right\|_2^2$. NSC classifies $\boldsymbol{z}$ to the class with the minimal representation residual. Let

$$\boldsymbol{W}_k = \boldsymbol{X}_k (\boldsymbol{X}_k^T \boldsymbol{X}_k + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}_k^T \tag{4.4}$$

The classification rule of NSC can be written as

$$label(\boldsymbol{z}) = \arg\min_k \|\boldsymbol{z} - \boldsymbol{W}_k\boldsymbol{z}\|_2^2 \tag{4.5}$$

Clearly, NSC learns a set of symmetric matrices $\boldsymbol{W}_k \in \Re^{d\times d}$ to reconstruct the query sample for classification.

4.2.2 Self-representation induced classifier

Representation based classifiers such as NSC, SRC and CRC rely on the similarity between samples. They assume that a query sample can be well represented by a linear combination of the training samples. Here we consider the representation based classification problem from a very different viewpoint. Considering the fact that the features of a sample are correlated (especially for visual data), we propose to represent each feature of a sample as the linear combination of all the features of this sample. Finally, the sample is represented by itself. Actually, such a self-representation strategy has been used successfully in image processing and feature selection [127]. For example, in image denoising a pixel (i.e., a feature) is represented as the weighted average of its neighboring pixels. In [127], feature similarity is defined and then representative features are selected by feature clustering.

Based on the above analysis, we present a self-representation based classification scheme. We can write the training subset of class k as $\boldsymbol{X}_k = \left[\boldsymbol{f}_{k1}; ...; \boldsymbol{f}_{kj}; ...; \boldsymbol{f}_{kd}\right]$ where $\boldsymbol{f}_{kj}$ is the j^{th} feature vector of $\boldsymbol{X}_k$. We represent $\boldsymbol{f}_{kj}$ as a linear combination of all the feature vectors:

$$\boldsymbol{f}_j^k = b_{j1} \times \boldsymbol{f}_{k1} +, ..., + b_{jd} \times \boldsymbol{f}_{kd} + \boldsymbol{e}_{kj} \tag{4.6}$$

where $b_{j1}, ..., b_{jd}$ are the representation coefficients and $\boldsymbol{e}_{jk}$ is the representation residual vector. Let $\boldsymbol{b}_j = [b_{j1}, ..., b_{jd}]$. Then Eq. (4.6) can be rewritten as $\boldsymbol{f}_{kj} = \boldsymbol{b}_j \boldsymbol{X}_k$. For all the feature vectors in $\boldsymbol{X}_k$, they can be represented by $\boldsymbol{X}_k$ with Eq. (4.6). Let $\boldsymbol{B}_k = [\boldsymbol{b}_1; \boldsymbol{b}_2; ...; \boldsymbol{b}_d]$ and $\boldsymbol{E}_k = [\boldsymbol{e}_1; \boldsymbol{e}_2; ...; \boldsymbol{e}_d]$. The representation of all features can be written as:

$$\boldsymbol{X}_k = \boldsymbol{B}_k \boldsymbol{X}_k + \boldsymbol{E}_k \tag{4.7}$$

We call the feature based representation model in Eq. (4.7) self-representation because it utilizes $\boldsymbol{X}_k$ to represent itself. To minimize the self-representation residual while avoiding the trivial solution, we have the following optimization problem:

$$\begin{aligned} &\min_{\boldsymbol{B}_k} l(\boldsymbol{E}_k) + R(\boldsymbol{B}_k) \\ &s.t. \boldsymbol{X}_k = \boldsymbol{B}_k \boldsymbol{X}_k + \boldsymbol{E}_k \end{aligned} \tag{4.8}$$

where $l(\boldsymbol{E}_k)$ is the loss function and $R(\boldsymbol{B}_k)$ is the regularization item. If we choose square loss and F-norm regularization, the problem in Eq. (4.8) becomes:

$$\hat{\boldsymbol{B}}_k = \arg\min_{\boldsymbol{B}_k} \|\boldsymbol{X}_k - \boldsymbol{B}_k \boldsymbol{X}_k\|_F^2 + \lambda \|\boldsymbol{B}_k\|_F^2 \tag{4.9}$$

Apparently, the problem in Eq. (4.9) has a closed-form solution:

$$\hat{\boldsymbol{B}}_k = \boldsymbol{X}_k \boldsymbol{X}_k^T \left(\boldsymbol{X}_k \boldsymbol{X}_k^T + \lambda \boldsymbol{I} \right)^{-1} \tag{4.10}$$

where $\boldsymbol{I} \in \Re^{d \times d}$ is an identity matrix. Given a query sample $\boldsymbol{z}$, its self-representation can then be computed as $\hat{\boldsymbol{B}}_k \boldsymbol{z}$ and the self-representation residual is $\boldsymbol{e} = \boldsymbol{z} - \hat{\boldsymbol{B}}_k \boldsymbol{z}$.

For each class, we can learn its self-representation matrix as above, and then we have a set of K self-representation matrices, $\boldsymbol{B}_1, ..., \boldsymbol{B}_k, ..., \boldsymbol{B}_K$ (we omit the superscript " ˆ " for the convenience of expression). The query sample $\boldsymbol{z}$ can be

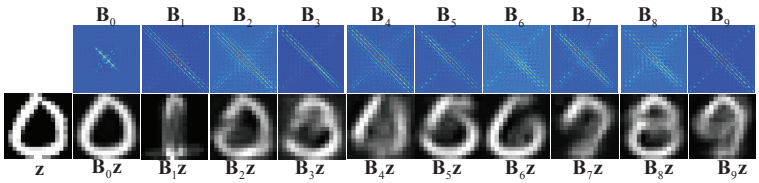


Figure 4.1 Top row: self-representation matrices $\boldsymbol{B}_k, k = 0, 1, ..., 9$ learned from the USPS database [85]. Bottom row: a query sample (from class 0) and its reconstructed images $\boldsymbol{B}_k\boldsymbol{z}, k = 0, 1, ..., 9$.

represented by each of the matrices and the classification can be made by checking which class has the minimal self-representation residual:

$$label(\boldsymbol{z}) = \arg\min_k \|\boldsymbol{z} - \boldsymbol{B}_k\boldsymbol{z}\|_2^2 \tag{4.11}$$

We call the above classifier self-representation induced classifier (SRIC).

We use an example to illustrate how SRIC works. As shown in Fig. 4.1, 10 self-representation matrices $\boldsymbol{B}_k, i = 0, 1, ..., 9$, are learned from handwritten digit dataset USPS [85]. Certainly, matrix $\boldsymbol{B}_k$ tends to represent better the features of sample from class k. Fig. 4.1 also shows a query sample $\boldsymbol{z}$ (from class 0) and the reconstructed samples $\boldsymbol{B}_k\boldsymbol{z}$ by all $\boldsymbol{B}_k$. We can see that $\boldsymbol{z}$ is well represented by $\boldsymbol{B}_0$ and it has the minimal self-representation residual on class 0, resulting in a correct classification.

4.2.3 Equivalence between SRIC and NSC

The NSC represents a sample from the perspective of sample similarity, while the proposed SRIC represents a sample from the perspective of feature similarity. Though the representation strategies are different, interestingly, it can be proved

that they lead to the same classification result. We have the following theorem.

Theorem 1 *SRIC is equivalent to l_2-norm regularized nearest subspace classifier, i.e., $\boldsymbol{B}_k$=$\boldsymbol{W}_k$, $k = 1, 2, ..., K$.*

Proof 1 *Applying singular value decomposition to $\boldsymbol{X}_k$, $\boldsymbol{X}_k = \boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T$, where $\boldsymbol{U}_k \in \Re^{d\times d}$, $\boldsymbol{\Lambda}_k \in \Re^{d\times n}$ and $\boldsymbol{V}_k \in \Re^{n\times n}$. Then $\boldsymbol{B}_k$ and $\boldsymbol{W}_k$ becomes:*

$$\begin{aligned}\boldsymbol{W}_k &= \boldsymbol{X}_i(\boldsymbol{X}_k^T\boldsymbol{X}_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}_k^T \\ &= \boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T(\boldsymbol{V}_k\boldsymbol{\Lambda}_k^T\boldsymbol{U}_k^T\boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{V}_k\boldsymbol{\Lambda}_k^T\boldsymbol{U}_k^T \\ &= \boldsymbol{U}_k\boldsymbol{\Lambda}_k(\boldsymbol{\Lambda}_k^T\boldsymbol{\Lambda}_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Lambda}_k^T\boldsymbol{U}_k^T\end{aligned} \tag{4.12}$$

$$\begin{aligned}\boldsymbol{B}_k &= \boldsymbol{X}_k\boldsymbol{X}_k^T\left(\boldsymbol{X}_k\boldsymbol{X}_k^T + \lambda\boldsymbol{I}\right)^{-1} \\ &= \boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T\boldsymbol{V}_k\boldsymbol{\Lambda}_k^T\boldsymbol{U}_k^T(\boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{V}_k^T\boldsymbol{V}_k\boldsymbol{\Lambda}_k^T\boldsymbol{U}_k^T + +\lambda\boldsymbol{I})^{-1} \\ &= \boldsymbol{U}_k\boldsymbol{\Lambda}_k\boldsymbol{\Lambda}_k^T(\boldsymbol{\Lambda}_k\boldsymbol{\Lambda}_k^T + +\lambda\boldsymbol{I})^{-1}\boldsymbol{U}_k^T\end{aligned} \tag{4.13}$$

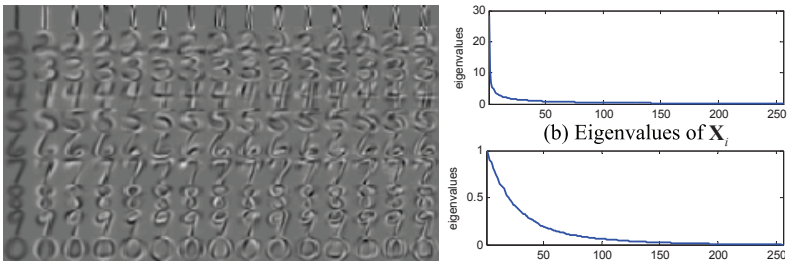
If $d < n$, we let $\boldsymbol{\Lambda}_k = [\boldsymbol{H}_k\ \boldsymbol{0}]$, where $\boldsymbol{H}_k \in \Re^{d\times d}$. Then we have

$$\boldsymbol{\Lambda}_b = \boldsymbol{\Lambda}_k(\boldsymbol{\Lambda}_k^T\boldsymbol{\Lambda}_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{\Lambda}_k^T = \boldsymbol{H}_k(\boldsymbol{H}_i^T\boldsymbol{H}_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{H}_k^T \tag{4.14}$$

$$\boldsymbol{\Lambda}_w = \boldsymbol{\Lambda}_k\boldsymbol{\Lambda}_k^T(\boldsymbol{\Lambda}_k\boldsymbol{\Lambda}_k^T + +\lambda\boldsymbol{I})^{-1} = \boldsymbol{H}_k\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{H}_k^T + \lambda\boldsymbol{I})^{-1} \tag{4.15}$$

Because $\boldsymbol{H}_k$ is a diagonal matrix, we have $\boldsymbol{\Lambda}_b = \boldsymbol{\Lambda}_w$. As $\boldsymbol{W}_k = \boldsymbol{U}_k\boldsymbol{\Lambda}_w\boldsymbol{U}_k^T$ and $\boldsymbol{B}_k = \boldsymbol{U}_k\boldsymbol{\Lambda}_b\boldsymbol{U}_k^T$, we can get $\boldsymbol{B}_k = \boldsymbol{W}_k$.

If $d > n$, $\boldsymbol{\Lambda}_k = \begin{bmatrix} \boldsymbol{H}_k \\ \boldsymbol{0} \end{bmatrix}$, where $\boldsymbol{H}_k \in \Re^{n\times n}$. $\boldsymbol{\Lambda}_b = \begin{pmatrix} \boldsymbol{H}_k(\boldsymbol{H}_k^T\mathbf{H}_k + \lambda\boldsymbol{I})^{-1}\boldsymbol{H}_k^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}$ and $\boldsymbol{\Lambda}_w = \begin{pmatrix} \boldsymbol{H}_k\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{H}_k^T + \lambda\boldsymbol{I})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}$. In this case, we can have the same conclusion, i.e., $\boldsymbol{\Lambda}_b = \boldsymbol{\Lambda}_w$ and $\boldsymbol{B}_k = \boldsymbol{W}_k$.



(a) Eigenvectors of $\mathbf{B}_i$ and $\mathbf{X}_i$ (b) Eigenvalues of $\mathbf{X}_i$ (c) Eigenvalues of $\mathbf{B}_i$

Figure 4.2 (a) The first 15 principle components of $\boldsymbol{B}_k$ and $\boldsymbol{X}_k$, $k = 0, 1, ..., 9$; (b) eigenvalues of $\boldsymbol{X}_k$; (b) eigenvalues of $\boldsymbol{B}_k$.

If $d = n$, let $\boldsymbol{\Lambda}_k = \boldsymbol{H}_k$, $\boldsymbol{\Lambda}_b$ and $\boldsymbol{\Lambda}_w$ are the same as those $\boldsymbol{\Lambda}_b$ and $\boldsymbol{\Lambda}_w$ when $d < n$. Hence, $\boldsymbol{B}_k = \boldsymbol{W}_k$ also holds on when $d = n$. ∎

From the above proof, we can have the following remark.

Remark 1 *SRIC is equivalent to principle component analysis with shrinkage.*

From the Proof, we can see that, $\boldsymbol{X}_k$ and $\boldsymbol{B}_k$ have the same set of eigenvectors, i.e., $\boldsymbol{U}_k$. Denote the h^{th} eigenvalue of $\boldsymbol{X}_k$ as σ_h, then the h^{th} eigenvalue $\boldsymbol{\Lambda}_{bh}$ of $\boldsymbol{B}_k$ will be $\frac{\sigma_h^2}{\lambda+\sigma_h^2}$. Therefore, for SIRC the eigenvalues of $\boldsymbol{B}_k$ will be shrunk to the range [0 1). The smaller the eigenvalue, the less the shrinkage ratio. Fig. 4.2(a) illustrates the first 15 principle components of $\boldsymbol{B}_k$ and $\boldsymbol{X}_k$ (please refer to Fig. 4.1 for $\boldsymbol{B}_k$). Fig. 4.2(b) and Fig. 4.2(C) plot the eigenvalues of $\boldsymbol{X}_k$ and $\boldsymbol{B}_k$, respectively. One can see that for the principle component of $\boldsymbol{X}_k$ with the largest eigenvalue, the corresponding eigenvalue of $\boldsymbol{B}_k$ is shrunk to nearly 1.

4.3 Discriminative self-representation induced classifier

4.3.1 Discriminative self-representation

The learning of self-representation matrix $\boldsymbol{B}_k$ in SRIC is rather generative but not discriminative since it only depends on the training data of class k. In light of the principle of self-representation in SRIC, we can then propose a discriminative self-representation induced classifier (DSRIC), which exploits the training data from all classes to learn $\boldsymbol{B}_k$.

SRIC aims to learn a $\boldsymbol{B}_k$ such that the self-representation residual $||\boldsymbol{X}_k - \boldsymbol{B}_k\boldsymbol{X}_k||_F^2$ could be minimized. However, SRIC does not take the samples of other classes into account. In order to make the classification more discriminative, we also expect that $\boldsymbol{B}_k$ cannot well represent the features of other classes. One may consider to maximize $\left\|\boldsymbol{X}_j - \boldsymbol{B}_k\boldsymbol{X}_j\right\|_F^2$, $j \neq k$ while minimizing $||\boldsymbol{X}_k - \boldsymbol{B}_k\boldsymbol{X}_k||_F^2$. However, this will make the whole objective function non-convex. Another much easier but still very reasonable choice is to learn a $\boldsymbol{B}_k$ such that the self-representation of $\boldsymbol{X}_j$, $j \neq k$, over it will approach to zero, i.e., $\left\|\boldsymbol{B}_k\boldsymbol{X}_j\right\|_F^2$ is very small. In other words, $\boldsymbol{B}_k$ is discriminative to represent the features of class k but not other classes. With these considerations, we propose the following DSRIC model to learn $\boldsymbol{B}_k$:

$$\hat{\boldsymbol{B}}_k = \arg\min_{\boldsymbol{B}_k} ||\boldsymbol{X}_k - \boldsymbol{B}_k\boldsymbol{X}_k||_F^2 + \lambda_1 \sum_{j\neq k} \left\|\boldsymbol{B}_k\boldsymbol{X}_j\right\|_F^2 + \lambda_2 ||\boldsymbol{B}_k||_F^2 \tag{4.16}$$

where λ_1 and λ_2 are the regularization parameters.

In Eq. (4.16), the first term $||\boldsymbol{X}_k - \boldsymbol{B}_k\boldsymbol{X}_k||_F^2$ aims to minimize the self-representation residual; the second term $\sum_{j\neq k} \left\|\boldsymbol{B}_k\boldsymbol{X}_j\right\|_F^2$ enforces that $\boldsymbol{X}_j$, $j \neq k$ will not be well rep-

resented by $\boldsymbol{B}_k$; the last term regularizes $\boldsymbol{B}_k$ to make the solution more stable. It is apparent that we still have a closed form solution of $\boldsymbol{B}_k$:

$$\hat{\boldsymbol{B}}_k = \boldsymbol{X}_k\boldsymbol{X}_k^T(\boldsymbol{X}_k\boldsymbol{X}_k^T + \lambda_1 \sum\nolimits_{j\neq k} \boldsymbol{X}_j\boldsymbol{X}_j^T + \lambda_2\boldsymbol{I})^{-1} \tag{4.17}$$

As shown in Fig. 4.3, we use a subset of AR database to show the difference between SRIC and DSRIC. Fig. 4.3(a) shows the query sample that belongs to subject 10. In Fig. 4.3(b), the query face $\boldsymbol{z}$ is well reconstructed by $\boldsymbol{B}_{10}$ learned by SRIC. However, from Fig. 4.3(d), we can see that $\boldsymbol{z}$ is misclassified to subject 15. The reconstructed faces using DSRIC are shown in Fig. 4.3(c). From Fig. 4.3(e), we can see that $\boldsymbol{z}$ is correctly classified to subject 10. Though the reconstruction ability of SRIC is superior to DSRIC, DSRIC has better discrimination ability than SRIC.

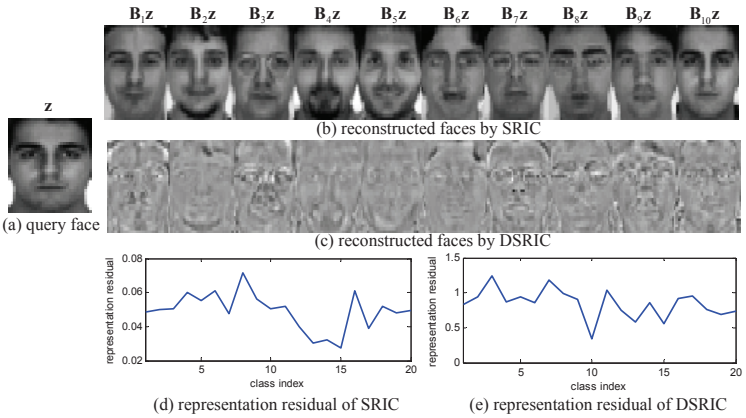


Figure 4.3 (a)query face $\boldsymbol{z}$; (b) reconstructed faces by SRIC; (c)reconstructed faces by DSRIC; (d) representation residual of each class (SRIC); (e) representation residual of each class (DSRIC).

Table 4.1 The algorithm of discriminative self-representation induced classifier (DSRIC).

Input: A query sample $\boldsymbol{z}$ and the training set $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_K]$.
Output: $label(\boldsymbol{z})$
1: Calculate $\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_K$ by Eq. (4.17);
2: Calculate $r_k = \|\boldsymbol{z} - \boldsymbol{B}_k\boldsymbol{z}\|_2^2$;
3: Get $label(\boldsymbol{z}) = \arg\min_k\{r_k\}$.

4.3.2 Classification and algorithms

After we get a set of matrices $\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_K$, a query sample $\boldsymbol{z}$ is classified to the class with the minimal reconstruction error.

$$label(\boldsymbol{z}) = \arg\min_k \|\boldsymbol{z} - \boldsymbol{B}_k\boldsymbol{z}\|_2^2 \tag{4.18}$$

The algorithm of DSRIC is shown in Table 4.1.

4.3.3 Complexity analysis

In this section, we discuss the time and space complexity of SRIC, DSRIC.

Training complexity

SRIC and DSRIC need to learn K self-representation matrices in the training stage by Eq. (4.10) and Eq. (4.17), respectively. The time complexity to solve Eq. (4.10) and Eq. (4.17) is $O(d^3)$. Hence the training time complexity of SRIC and DSRIC is $O(Kd^3)$. During the training stage, all the methods should contain the training set. Hence, the training memory of SRIC and DSRIC is $Kd^2 + Kdn$.

Testing complexity

In the testing stage, the time complexity of SRIC and DSRIC is $O(Kd^2)$. As D-SRIC only needs to store a set of $d \times d$ matrices, the storage space of DSRIC is Kd^2. When the number of samples is much larger than the number of feature dimensions, the advantage of DSRIC in time complexity and storage consumption is quite significant.

We will compare SRIC and DSRIC with NNC [38], SVM [24], NSC [31], NAH [186], NCH [186], SRC [201], CRC [226] and CROC [30] in the experiments. The time and space complexity in the training and testing stages of all the methods are listed in Table 4.2.

Table 4.2 Time complexity and memory consumption of different classifiers.

method	NNC [38]	SVM	SRC [201]	NSC [31]	SRIC
Time(train)	/	$O(Kdn)$	/	$O(Kn^3)$	$O(Kd^3)$
Time(test)	$O(Kdn)$	$O(Kd)$	$O(d^2n^\varepsilon)$	$O(Kdn)$	$O(Kd^2)$
Memory(train)	/	$Kd + Kdn$	/	$2Kdn + n^2$	$Kd^2 + Kdn$
Memory(test)	Kdn	Kd	Kdn	$2Kdn$	Kd^2
method	NCH [186]	NAH [186]	CRC [226]	CROC [30]	DSRIC
Time(train)	/	/	$O((Kn)^3)$	$O((Kn)^3 + Kn^3)$	$O(Kd^3)$
Time(test)	$O((Kn)^3)$	$O((Kn)^3)$	$O(Kdn)$	$O(Kdn)$	$O(Kd^2)$
Memory(train)	/	/	$2Kdn + (Kn)^2$	$3Kdn + (Kn)^2$	$Kd^2 + Kdn$
Memory(test)	Kdn	Kdn	$2Kdn$	$3Kdn$	Kd^2

4.4 Experimental analysis

In this section, we test the performance of DSRIC[1] on eight UCI datasets [6], two handwritten digit recognition databases [85][104], two face recognition database [59][82] and one gender classification dataset [123]. We compare the proposed classifier with eight popular and state-of-the-art classifiers, including the nearest neighbor classifier (NNC) [38], support vector machines (SVM) [24], nearest subspace classifier (NSC) [31], nearest convex hull classifier (NCH) [186], nearest affine hull classifier (NAH) [186], sparse representation based classifier (SRC) [201], collaborative representation based classifier (CRC) [226] and collaborative representation optimization classifier (CROC) [30]. Among them, NNC and SVM are baseline benchmarks, and the remaining are all representation based classifiers.

The performance of different classifiers is evaluated from three aspects: classification accuracy, the running time and memory consumption in the testing stage. In order to easily show the speedup and memory saving of DSRIC over other methods, in all the following experiments we take the running time and memory consumption of DSRIC as a unit (i.e., 1), and report the results of other methods based on it. All algorithms are run in an Intel(R) Core(TM) i7-2600K (3.4GHz) PC.

4.4.1 Parameter setting

There are two parameters in DSRIC: λ_1 and λ_2. In all the experiments, λ_2 is fixed as 0.001 and λ_1 is chosen on the training dataset by five-fold cross-validation. For the compared representation based methods, the parameters in NCH and NAH are

[1] Since SRIC is equivalent to NSC, the results of SRIC will not be reported.

set as 1 and 100, respectively, as suggested in the original paper; the regularization parameter in NSC, SRC and CRC is tuned from $\{0.0005, 0.001, 0.005, 0.01\}$ and the best results are reported; following the experiment setting in [30], the parameter of CROC is chosen by five-fold cross-validation on the training set.

4.4.2 UCI datasets

We first use eight datasets (derm, german, heart, hepatitis, iono, rice, thyroid, wdbc, wpbc, yeast) from the UCI machine learning repository [6] to evaluate the performance of DSRIC. The number of classes (c), number of features (f) and number of samples (s) of the eight datasets are illustrated in the right column of Table 4.3. The average classification accuracy, testing time and testing memory over the eights datasets are listed at the bottom of Table 4.3.

From Table 4.3, we can see that the accuracy of DSRIC is about 2% higher than NSC, SRC and CRC, and 3% higher than CROC. Besides, DSRIC is much faster than the other representation based classifiers. Compared with NSC, SRC, CRC and CROC, the running time speedup by DSRIC is 64, 547, 106 and 130, respectively. Because NAH and NCH have to solve a QP problem for each query sample, the time consumption is very high compared with other classifiers. In terms of memory requirement, in this experiment DSRIC also has clear advantage. It costs less than 1/10 memory of other classifiers except for SVM, which is not a representation based classifier.

Table 4.3 Classification accuracy, testing time and testing memory on UCI datasets.

Database	NNC	SVM	SRC	NSC	NCH	NAH	CRC	CROC	DSRIC	c/f/s
derm	96.1	96.5	97.1	97.4	96.5	96.9	97.1	97.7	97.6	6/34/366
german	68.8	73.6	74.1	70.6	70.6	71.3	72.9	72.9	73.4	2/20/1000
heart	76.7	83.3	83	78.1	76.3	76.5	84.1	83.3	83.7	2/13/270
hepatitis	82.5	86.2	86.8	86.8	82.1	81.7	84.7	86.7	87.5	2/19/155
iono	86.4	87.6	91	94.4	89.2	80.3	92.7	83.3	94.7	2/34/351
rice	80	78.2	82.9	84.7	80.7	80.5	83.8	82.9	86.6	2/5/104
thyroid	95.3	89.8	90.2	95.8	96.3	95.8	91.1	87.4	95.8	3/5/215
wdbc	95.4	97.7	93.5	92.3	94	93.9	94.7	95.3	95.6	2/30/569
wpbc	70.7	77.4	79.4	76.8	75.4	74.7	76.3	79.4	80.9	2/33/198
yeast	48.8	56.4	54.9	56.9	49.3	50.1	54.6	54.3	57.7	10/7/1484
Accuracy	80.1	82.7	83.3	83.4	81.0	80.2	83.2	82.3	**85.4**	
Time	2.8×10^4	1.4	547	64	4.9×10^5	6.6×10^5	106	130	**1**	
Memory	10.48	0.08	10.48	20.97	10.48	10.48	20.97	31.45	**1**	

4.4.3 Handwritten digit recognition

USPS

The USPS dataset contains 7,291 training and 2,007 testing images [85]. Each class has about 650 training samples, and each handwritten digit sample is a 16×16 image. The experimental results are listed in listed in Table 4.4. Since each class has enough training samples and the feature dimension is not high in this experiment,

the simple NNC achieves the best accuracy. The recognition rate of DSRIC is only 0.3% lower than NNC. However, DSRIC is significantly faster than NNC with 10,000 times speedup. In addition, the memory consumption of NNC is 2.8 times larger than DSRIC.

Table 4.4 Recognition rate, testing time and testing memory on USPS dataset.

Method	NNC	SVM	SRC	NSC	NCH	NAH	CRC	CROC	DSRIC
Accuracy	94.6	92.9	94.0	94.3	91.9	92.3	90.6	90.1	94.3
Time	1×10^4	22.9	1×10^4	165.6	5.1×10^4	7.7×10^4	150.8	977.1	**1**
Memory	2.848	0.004	2.848	5.696	2.848	2.848	5.696	8.544	**1**

MNIST

The MNIST [104] dataset includes a training set of 60,000 samples and a test set of 10,000 samples. The size of each image is 28×28 and there are 10 classes of digit images. Compared to USPS, there are more training samples. Table 4.5 lists the recognition rate, testing time and testing memory by different methods. Similar to the results in USPS, the recognition rate of DSRIC equals to NSC, and 1.4% lower than NNC. However, DSRIC avoids the one-to-one searching process in the training set and is 18,000 faster than NNC, which is very important in real-time applications. Compared with SRC, DSRIC is 51 times faster and saves 7.65 times the memory. Please note that the performances of NCH, NAH, CRC and CROC are not reported because these methods need to process a 60,000×60,000 square matrix and out-of-memory in our PC.

Table 4.5 Recognition rate, testing time and testing memory on MNIST dataset.

Method	NNC	SVM	SRC	NSC	NCH	NAH	CRC	CROC	DSRIC
Accuracy	97.1	94.6	94.5	95.7	/	/	/	/	95.7
Time	1.8×10^4	51	6.3×10^4	649.3	/	/	/	/	**1**
Memory	7.653	0.001	7.653	15.306	7.653	7.653	15.306	22.959	**1**

4.4.4 Face recognition

Extended Yale B database

The Extended Yale B database contains about 2,414 frontal face images of 38 individuals [59]. The face images were cropped and resized to 24×21 pixels. Following the experiment setting in [201][214], Subsets 1 and 2 (717 images, normal-to-moderate lighting conditions) are used for training, and Subset 3 (453 images, more extreme lighting conditions) is used for testing. The experimental results are shown in Table 4.6. From Table 4.6, we can see that DSRIC achieves the best recognition rate. Compared with SRC, the FR efficiency is greatly improved.

Table 4.6 Recognition rate, testing time and testing memory on Extended Yale B database.

Method	NN	SVM	SRC	NSC	NCH	NAH	CRC	CROC	DSRIC
Accuracy	47.0	92.5	97.6	78.4	67.7	80.1	97.2	97.1	**97.8**
Time	68.8	0.7989	40.1	1.2	230.1	90.1	1.5	3.5	1
Memory	0.0374	0.002	0.0374	0.0749	0.0374	0.0374	0.0749	0.1123	1

LFW database

The LFW database [82] contains images of 5,749 subjects in unconstrained environment. LFW-a is a version of LFW after alignment using commercial face alignment software. We gathered the subjects which have no less than eleven samples and then formed a dataset with 136 subjects from LFW-a. Each face image is firstly cropped to 102×120 and then resized to 32×32 images. Some face images of LFW database are shown in Fig. 4.4. We select 9 face images per subject for training and use the remaining face images for testing. Hence, there are 1,224 training samples and the feature dimension is 1024.

Figure 4.4 Face images of LFW database.

The experimental results are shown in Table 4.7. Though SVM has the fastest speed and least memory requirement, it has the worse accuracy. The representation based classifiers all lead to much better accuracy than SVM. DSRIC has the highest recognition accuracy. Since there are 158 subject and the feature dimension is 1024, DSRIC does not show advantages in memory in this experiment.

4.4.5 Gender classification

In this section, a non-occluded subset (14 images per subject) of the AR dataset [123] is used. It includes face images of 50 male and 50 female subjects. The images from the first 25 males and 25 females are used for training and the remaining

Table 4.7 Recognition rate, testing time and testing memory on LFW database.

Method	NNC	SVM	SRC	NSC	NCH	NAH	CRC	CROC	DSRIC
Accuracy	20.1	16.3	60.4	37.8	34.5	37.7	58.8	60.0	**60.8**
Time	14.7	0.32	25	0.55	80.2	107.9	0.77	1.28	**1**
Memory	0.009	0.001	0.009	0.018	0.009	0.009	0.018	0.026	1

for testing. Following the experiment setting in [226], each face image is cropped to 60×43 and PCA is used to reduce the feature dimension to 50. The classification accuracy, testing time and testing memory are given in Table 4.8. One can see that DSRIC achieves the highest accuracy, and it costs much less running time and memory than others (except for SVM in memory consumption).

Table 4.8 Classification accuracy, testing time and testing memory on on Gender classification dataset.

Method	NNC	SVM	SRC	NSC	NCH	NAH	CRC	CROC	DSRIC
Accuracy	90.3	91.4	93.1	93.4	91.4	91.4	93.1	92.9	**94.7**
Time	1.4×10^4	23.8	8.4×10^3	44.4	2.5×10^5	3.6×10^5	41.1	92	**1**
Memory	7	0.2	7	14	7	7	14	21	**1**

4.5 Conclusions and discussions

In this chapter we investigated the representation based classification problem from a "feature oriented" perspective. Different from the existing representation based classifiers that represent a sample as the linear combination of other samples, we

explored to represent a feature by its relevant features in the data, which we call self-representation. A self-representation induced classifier (SRIC) was then proposed, which learns a self-representation matrix per class and uses these matrices for classification. The query sample is then classified to the class with the minimal reconstruction error. We proved that SRIC is equivalent to nearest subspace classifier (NSC) with l_2-norm regularization in terms of classification decision. Furthermore, it can be shown that SRIC is essentially the principle component analysis (PCA) with eigenvalue shrinkage. We then proposed a discriminative SRIC (D-SRIC) classifier, which not only minimizes the feature self-representation residual of this class but represents little the features of other classes. The time and space complexity of DSRIC (except for the training memory) is invariant to the number of training samples, which makes it very suitable for large scale datasets with many training samples, e.g., USPS and MNIST. Experimental results on different pattern recognition tasks showed that DSRIC achieves comparable or superior recognition rate to state-of-the-art representation based classifiers, while it has higher efficiency and lower memory consumption.

As the time complexity of the proposed DSRIC is only related with the number of features and classes, DSRIC can well apply to classification with large amounts of samples. However, in computer vision tasks, the image features are usually high-dimensional. In this case, we can reduce the feature dimension firstly and then use DSRIC for classification.

Chapter 5

Image Set based Collaborative Representation

Apart from image based classification, in practice there are also many image set based classification problems, e.g., video based face recognition, multi-view object recognition. Intuitively, representation based classifiers (i.e., SRC/CRC) can be directly extended to image set based classification tasks by representing each image of the set separately. However, they ignore the distinctiveness of samples in the query image set. The existing set to set distances ignore the correlation among the training image sets. Additionally, the redundancy in the image set should be taken into account. In this chapter, we develop image set based collaborative representation models, which simultaneously consider the distinctiveness of samples in the query image set, the correlation among the training image sets and the redundancy in the image set.

5.1 Introduction

Image set based classification has been increasingly employed in face recognition [5, 21, 29, 40, 78, 136, 145, 193, 199, 206] and object categorization [98, 190] in recent years. Due to the rapid development of digital imaging and communication techniques, now image sets can be easily collected from multi-view images using multiple cameras [98], long term observations [199], personal albums and news pictures [162], etc. Meanwhile, image set based face recognition (ISFR) has shown superior performance to single image based face recognition since the many sample images in the gallery set can convey more within-class variations of the subject [78]. One special case of ISFR is video based face recognition, which collects face image sets from consecutive video sequences [105, 171, 206]. Similar to the work in [21, 78], in this chapter we focus on the general case of ISFR without considering the temporal relationship of samples in each set.

The key issues in image set based classification include how to model a set and consequently how to compute the distance/similarity between query and gallery sets. Researchers have proposed parametric and non-parametric approaches for image set modeling. Parametric modeling methods model each set as a parametric distribution, and use Kullback-Leibler divergence to measure the similarity between the distributions [5, 199]. The disadvantage of parametric set modeling lies in the difficulty of parameter estimation, and it may fail when the estimated parametric model does not fit well the real gallery and query sets [78, 98, 193].

Many non-parametric set modeling methods have also been proposed, including subspace [98, 206], manifold [47, 69, 190, 191, 193], affine hull [21, 78], convex hull [21], and covariance matrix based ones [20, 87, 191]. The method in [98]

employs canonical correlation to measure the similarity between two sets. A projection matrix is learned by maximizing the canonical correlations of within-class sets while minimizing the canonical correlations of between-class sets. The methods in [192] use manifold to model an image set and define a manifold-to-manifold distance (MMD) for set matching. MMD models each image set as a set of local subspaces and the distance between two image sets is defined as a weighted average of pairwise subspace to subspace distance. As MMD is a non-discriminative measure, Manifold Discriminant Analysis (MDA) is proposed to learn an embedding space by maximizing manifold margin [190]. The performance of subspace and manifold based methods may degrade much when the set has a small sample size but big data variations [78, 191]. In affine hull and convex hull based methods [21, 78], the between-set distance is defined as the distance between the two closest points of the two sets. When convex hull is used, the set to set distance is equivalent to the nearest point problem in SVM [18]. In [79], a method called sparse approximated nearest points (SANP) is proposed to measure the dissimilarity between two image sets. To reduce the model complexity of SANP, a reduced model, which is called regularized nearest points (RNP), is proposed by modeling each image set as a regularized hull [220]. However, the closest points based methods [21, 78, 204, 220] rely highly on the location of each individual sample in the set, and the model fitting can be heavily deteriorated by outliers [191]. A collaborative regularized nearest points (CRNP) method is proposed in [203] to extend RNP.

To improve the classification performance, the kernel trick can be introduced to map the image sets to high-dimensional subspaces, e.g., kernel mutual subspace method [51] and kernel discriminant transformation [32]. In [191], an image set is

represented by a covariance matrix and a Riemannian kernel function is defined to measure the similarity between two image sets by a mapping from the Riemannian manifold to a Euclidean space. With the kernel function between two image sets, traditional discriminant learning methods, e.g., linear discriminative analysis [10], partial least squares [154], kernel machines, can be used for image set classification [20, 87]. The disadvantages of covariance matrix based methods include the computational complexity of eigen-decomposition of symmetric positive-definite (SPD) matrices and the curse of dimensionality with limited number of training sets.

No matter how the set is modeled, in almost all the previous works [21, 47, 69, 78, 98, 190, 191, 193, 206, 220], the query set is compared to each of the gallery sets separately, and then classified to the class closest to it. Such a classification scheme does not consider the correlation between gallery sets, like the nearest neighbor or nearest subspace classifier in single image based face recognition. In recent years, the sparse representation based classification (SRC) [201] has shown interesting results in image based face recognition. SRC represents a query face as a sparse linear combination of samples from all classes, and classifies it to the class which has the minimal representation residual to it. Though SRC emphasizes much on the role of l_1-norm sparsity of representation coefficients, it has been shown in [226] that the collaborative representation mechanism (i.e., using samples from all classes to collaboratively represent the query image) is more important to the success of SRC. The so-called collaborative representation based classification (CRC) with l_2-regularization leads to similar results to SRC but with much lower computational cost [226]. In [217], feature weights are introduced to the representation model to penalize pixels with large error so that the model is robust to outliers. Moreover,

a kernel sparse representation model is proposed for face recognition by mapping features to a high dimensional Reproducing Kernel Hilbert Space (RKHS), which further improves the recognition accuracy [56]. Similarly, a robust kernel representation model is proposed with iteratively reweighted algorithms [216].

One may apply SRC/CRC to ISFR by representing each image of the query set over all the gallery sets, and then using the average or minimal representation residual of the query set images for classification. However, such a scheme does not exploit the correlation and distinctiveness of sample images in the query set. If the average representation residual is used for classification, the discrimination of representation residuals by different classes will be reduced; if the minimal representation residual is used, the classification can suffer from the outlier images in the query set. In addition, there are redundancies in an image set. The redundancies will lead to great storage burden and computational complexity, and deteriorate the recognition performance.

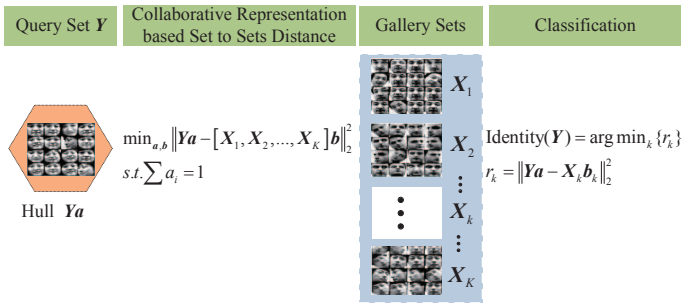


Figure 5.1 Image set based collaborative representation and classification (ISCRC).

In this chapter, we propose a novel image set based collaborative representation and classification (ISCRC) approach for ISFR, as illustrated in Fig. 5.1. The query set, denoted by $\boldsymbol{Y}$ (each column of $\boldsymbol{Y}$ is an image in the set) is modeled as a hull

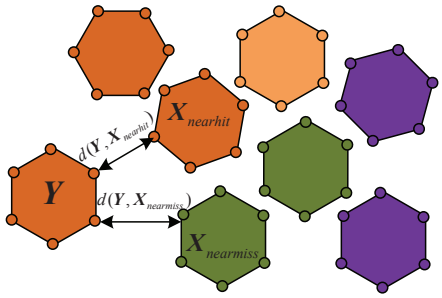


Figure 5.2 Illustration of image set margin.

$\boldsymbol{Ya}$ with the sum of coefficients in $\boldsymbol{a}$ being 1. Let $\boldsymbol{X}_k, k = 1, 2, ..., K$, be a gallery set. We then propose a collaborative representation based set (i.e., $\boldsymbol{Y}$) to sets (i.e., $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_k, ..., \boldsymbol{X}_K]$) distance (CRSSD for short); that is, we represent the hull $\boldsymbol{Ya}$ over the gallery sets $\boldsymbol{X}$ as $\boldsymbol{Xb}$, where $\boldsymbol{b}$ is a coefficient vector. Consequently, we can classify the query set $\boldsymbol{Y}$ by checking which gallery set has the minimal representation residual to the hull $\boldsymbol{Ya}$. To get a stable solution to CRSSD, regularizations can be imposed on $\boldsymbol{a}$ and $\boldsymbol{b}$. In the proposed ISCRC, the gallery sets $\boldsymbol{X}_k$ can be compressed to a smaller size to remove the redundancy so that the time complexity of ISCRC can be much reduced without sacrificing the recognition rate. Our experiments on three benchmark ISFR databases show that the proposed ISCRC is superior to state-of-the-art methods in terms of both recognition rate and efficiency.

To better illustrate the motivation of ISCRC, we use an example to explain the superiority of ISCRC over set to set distance based classifiers (e.g., CHISD [21], SANP [78], RNP [220]) from a large margin perspective. Large margin principle has been widely used in classifier design (e.g., SVM [18], LVQ [39]), ensemble learning (e.g., AdaBoost [155]) and metric learning (e.g., MDA [190], LMNN [14]). In classification, large margin can lead to better generalization ability [167].

In [195], SRC is interpreted as a margin classifier and a margin is derived for SRC. Actually, in image set based classification, MDA [190], DCC [98] and CDL [191] all try to learn a discriminative set to set distance in a large margin manner, i.e., pull the similar image sets together while push the dissimilar image sets away. Similar to sample margin in nearest neighbor classifier, image set margin can be defined. Given a query set $\boldsymbol{Y}$ but multiple gallery sets $\boldsymbol{X}_k, k = 1, 2, ..., K$, as illustrated in Fig. 5.2, the image set margin is defined as:

$$margin_{\boldsymbol{Y}} = d(\boldsymbol{Y}, \boldsymbol{X}_{nearmiss}) - d(\boldsymbol{Y}, \boldsymbol{X}_{nearhit}) \tag{5.1}$$

where $\boldsymbol{X}_{nearhit}$ is the nearest gallery set of $\boldsymbol{Y}$ with the same class label, $\boldsymbol{X}_{nearmiss}$ is the nearest gallery set of $\boldsymbol{Y}$ with a different class label, $d(\boldsymbol{Y}, \boldsymbol{X}_{nearmiss})$ is the distance between $\boldsymbol{Y}$ and $\boldsymbol{X}_{nearmiss}$, and $d(\boldsymbol{Y}, \boldsymbol{X}_{nearhit})$ is the distance between $\boldsymbol{Y}$ and $\boldsymbol{X}_{nearhit}$. If $margin_{\boldsymbol{Y}}$ is positive, $\boldsymbol{Y}$ can be correctly classified; otherwise, $\boldsymbol{Y}$ would be misclassified. Hence, a large margin is desired in image set classification.

Fig. 5.3 shows the margin comparison between the proposed ISCRC and hull based set to set distances (i.e., CHISD [21] and RNP [220]), where the Honda/USCD[1] database [105] is used. Fig. 5.3(a) is the comparison between ISCRC and convex hull based image set distance, i.e., CHISD. The image sets marked by pentagram are misclassified by CHISD with negative margin while correctly classified by ISCRC with positive margin. Besides, the margin of the other image sets are all enlarged, which represents better generalization ability in classification. Fig. 5.3(b) illustrates the comparison between ISCRC and regularized hull based image set distance, i.e., RNP. Although RNP classifies all the image sets correctly with positive margin, ISCRC results in much larger margin than RNP. Both comparisons

[1] http://vision.ucsd.edu/ leekc/HondaUCSDVideoDatabase/HondaUCSD.html

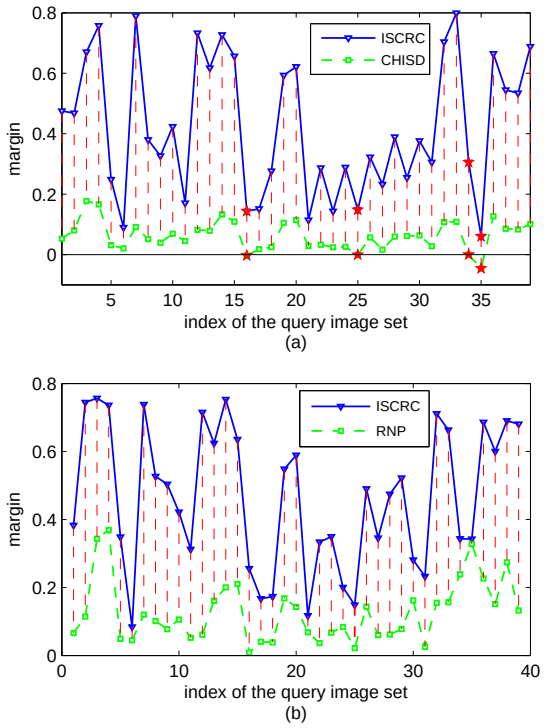


Figure 5.3 Margin comparison between ISCRC and CHISD (a) and RNP (b).

show that the proposed ISCRC can lead to larger image set margin compared with set to set distance, indicating that ISCRC would get better generalization performance.

The rest of this chapter is organized as follows. Section 5.2 discusses in detail the proposed CRSSD and ISCRC methods. Section 5.3 presents the regularized hull based ISCRC, followed by the convex hull based ISCRC in Section 5.4. Section 5.5 conducts experiments and Section 5.6 gives our conclusions. The main abbreviations used in the development of our method are summarized in Table 5.1.

Table 5.1 The main abbreviations used in this chapter.

ISFR	image set based face recognition
SRC	sparse representation based classification
CRC	collaborative representation based classification
CRSSD	collaborative representation based set to sets distance
ISCRC	image set based collaborative representation and classification
RH-ISCRC	regularized hull based ISCRC
KCH-ISCRC	kernelized convex hull based ISCRC

5.2 Collaborative representation based set to sets distance

We first introduce the hull based set to set distance in 5.2.1, and then propose the collaborative representation based set to sets distance (CRSSD) in 5.2.2. With CRSSD, the image set based collaborative representation and classification (ISCRC) scheme can be naturally proposed. In 5.2.3 and 5.2.4, the convex hull and regularized hull based CRSSD are respectively presented.

5.2.1 Hull based set to set distance

In image set based classification, compared to the parametric modeling of image set, non-parametric modeling does not impose assumptions on the data distribution and inherits many favorable properties [78, 98, 191]. One simple non-parametric

set modeling approach is the hull based modeling [21, 78], which models a set as the linear combination of its samples. Given a sample set $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_i, ..., \boldsymbol{y}_{n_a}\}$, $\boldsymbol{y}_i \in \Re^d$, the hull of set $\boldsymbol{Y}$ is defined as: $H(\boldsymbol{Y}) = \{\sum a_i \boldsymbol{y}_i\}$. Usually, $\sum a_i = 1$ is required and the coefficients a_i are required to be bounded:

$$H(\boldsymbol{Y}) = \{\sum a_i \boldsymbol{y}_i \,|\, \sum a_i = 1, 0 \leq a_i \leq \tau\} \tag{5.2}$$

If $\tau = 1$, $H(\boldsymbol{Y})$ is a convex hull [153]. If $\tau < 1$, $H(\boldsymbol{Y})$ is a reduced convex hull [18]. For the convenience of expression, in the following development we call both the cases convex hull.

By modeling a set as a convex hull, the distance between set $\boldsymbol{Y} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_i, ..., \boldsymbol{y}_{n_a}\}$ and set $\boldsymbol{Z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_j, ..., \boldsymbol{z}_{n_z}\}$ can be defined as follows:

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{b}} \left\| \sum a_i \boldsymbol{y}_i - \sum b_j \boldsymbol{z}_j \right\|_2^2 \\ &s.t. \sum a_i = 1, 0 \leq a_i \leq \tau \\ &\sum b_j = 1, 0 \leq b_j \leq \tau \end{aligned} \tag{5.3}$$

When the two sets have no intersection, the set to set distance in Eq. (5.3) becomes the distance between the nearest points in the two convex hulls (CHISD [21]), as illustrated in Fig. 5.4. It is not difficult to see that such a distance is equivalent to the distance computed by SVM [18]. If the discriminative function of SVM is $f = \boldsymbol{w}\boldsymbol{x} + b$, then $\boldsymbol{w} = \sum a_i \boldsymbol{y}_i - \sum b_j \boldsymbol{z}_j$ and the margin is $2/\|\boldsymbol{w}\|$. If we consider each image set as one class, then maximizing margin between the two classes is equivalent to finding the set to set distance [23]. In image set based face recognition, there is usually no intersection between image sets of different persons. If there are intersections between two image sets, then τ can be set as below 1 and the resulting problem can be related with soft-margin SVM and υ-SVM [12, 21]. Unfortunately,

such a distance relies highly on the location of each individual sample and can be sensitive to outliers [191]. More detailed discussions about convex/affine hull based classifiers can be found in [12, 18, 21, 142].

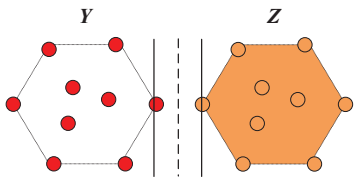


Figure 5.4 Convex hull based set to set distance.

5.2.2 Collaborative representation based set to sets distance and classification

In image set based face recognition (ISFR), we have a query set $\boldsymbol{Y}$ but multiple gallery sets $\boldsymbol{X}_k, k = 1, 2, ..., K$. One fact in face recognition is that the face images from different people still have much similarity. If we compute the distance between $\boldsymbol{Y}$ and each $\boldsymbol{X}_k$ by using methods such as hull based set to set distance (refer to 6.2.1), the correlation between different gallery sets will not be utilized. As we discussed in the Introduction section, inspired by the SRC [201] and CRC [226] methods in image based face recognition, here we propose a novel ISFR method, namely image set based collaborative representation and classification (ISCRC).

The key component of ISCRC is the collaborative representation based set to sets distance (CRSSD) defined as follows. Let $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_k, ..., \boldsymbol{X}_K]$ be the concatenation of all gallery sets. We model each of $\boldsymbol{Y}$ and $\boldsymbol{X}$ as a hull, i.e., $\boldsymbol{Ya}$ and $\boldsymbol{Xb}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are coefficient vectors, and then we define the CRSSD between set

$\boldsymbol{Y}$ and sets $\boldsymbol{X}$ as:

$$\min_{\boldsymbol{a},\boldsymbol{b}} \|\boldsymbol{Y}\boldsymbol{a} - \boldsymbol{X}\boldsymbol{b}\|^2 \;\; s.t. \sum a_i = 1 \tag{5.4}$$

where a_i is the i^{th} coefficeint in $\boldsymbol{a}$ and we let $\sum a_i = 1$ to avoid the trivial solution $\boldsymbol{a} = \boldsymbol{b} = \boldsymbol{0}$. In Eq. (5.4), the hull $\boldsymbol{Y}\boldsymbol{a}$ of the query set $\boldsymbol{Y}$ is collaboratively represented over the gallery sets; however, the coefficients in $\boldsymbol{a}$ will make the samples in $\boldsymbol{Y}$ be treated differently in the representation and the subsequent classification process. By minimizing the distance between $\boldsymbol{Y}\boldsymbol{a}$ and $\boldsymbol{X}\boldsymbol{b}$, the outliers (e.g., one frame with large corruptions/occlusions) in both the query image set $\boldsymbol{Y}$ and the gallery image sets $\boldsymbol{X}$ will be assigned with very small representation coefficients. Therefore, the impact of outliers can be much alleviated. Our experimental results in Section 6.4 showed that ISCRC is robust to face variations in different conditions.

Suppose that the coefficient vectors $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{b}}$ are obtained by solving Eq. (5.4), then we can write $\hat{\boldsymbol{b}}$ as $\hat{\boldsymbol{b}} = [\hat{\boldsymbol{b}}_1; ...; \hat{\boldsymbol{b}}_k; ...; \hat{\boldsymbol{b}}_K]$, where $\hat{\boldsymbol{b}}_k$ is is the sub-vector of coefficients associated with gallery set $\boldsymbol{X}_k$. Similar to the classification in SRC and CRC, we use the representation residual of hull $\boldsymbol{Y}\hat{\boldsymbol{a}}$ by each set $\boldsymbol{X}_k$ to determine the class label of $\boldsymbol{Y}$. The classifier in the proposed ISCRC is:

$$Identity(\boldsymbol{Y}) = argmin_k \{r_k\} \tag{5.5}$$

where $r_k = \left\|\boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{X}_k\hat{\boldsymbol{b}}_k\right\|_2^2$.

Clearly, the solutions to $\boldsymbol{a}$ and $\boldsymbol{b}$ in Eq. (5.4) determine the CRSSD and hence the result of ISCRC. In order to get stable solutions, we could impose reasonable regularizations on $\boldsymbol{a}$ and $\boldsymbol{b}$. In the following sections 6.2.3 and **??**, we discuss the convex hull based CRSSD and regularized hull based CRSSD, respectively.

5.2.3 Convex hull based CRSSD

One important instantiation of CRSSD is the convex hull based CRSSD. In this case, both the hulls $\boldsymbol{Ya}$ and $\boldsymbol{Xb}$ are required to be convex hulls, and then the distance in Eq. (5.4) becomes

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{b}} \|\boldsymbol{Ya} - \boldsymbol{Xb}\|^2 \\ &s.t. \sum a_i = 1, \sum b_j = 1, \\ &\quad 0 \le a_i \le \tau, i = 1, ..., n_a, \\ &\quad 0 \le b_j \le \tau, j = 1, ..., n_b \end{aligned} \tag{5.6}$$

where a_i and b_j are the i^{th} and j^{th} coefficients in $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively, n_a and n_b are the number of samples in set $\boldsymbol{Y}$ and sets $\boldsymbol{X}$, respectively, and $\tau \le 1$.

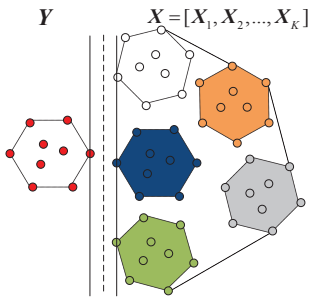


Figure 5.5 Convex hull based CRSSD.

A geometric illustration of convex hull based CRSSD is shown in Fig. 5.5. Different from the CHISD method in [21], which models each gallery set as a convex hull, here we model all the gallery sets as one big convex hull. Similar to the closest points searching in SVM, convex hull based CRSSD aims to find the closest points in the query set $\boldsymbol{Y}$ and the whole gallery set $\boldsymbol{X}$ in a large margin manner. With convex hull based CRSSD, the corresponding ISCRC method can be viewed as a large

margin based classifier in some sense. Nonetheless, the classification rules in SVM and ISCRC are very different.

5.2.4 l_p-norm regularized hull based CRSSD

The convex hull modeling of a set can be affected much by outlier samples in the set [191]. To make CRSSD more stable, the l_p-norm regularized hull can be used to model $\boldsymbol{Y}$ and $\boldsymbol{X}$. For the query set $\boldsymbol{Y}$, we should keep the constraint $\sum a_i = 1$ to avoid the trivial solution, and the l_p-norm regularized hull of $\boldsymbol{Y}$ is defined as

$$H(\boldsymbol{Y}) = \{\sum a_i \boldsymbol{y}_i \,|\, \|\boldsymbol{a}\|_{l_p} < \delta\} \; s.t. \sum a_i = 1 \tag{5.7}$$

For the gallery set $\boldsymbol{X}$, its regularized hull is defined as:

$$H(\boldsymbol{X}) = \{\sum b_i \boldsymbol{x}_i \,|\, \|\boldsymbol{b}\|_{l_p} < \delta\} \tag{5.8}$$

Finally, the regularized hull based CRSSD between $\boldsymbol{Y}$ and $\boldsymbol{X}$ is defined as:

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{b}} \|\boldsymbol{Y}\boldsymbol{a} - \boldsymbol{X}\boldsymbol{b}\|_2^2 \\ &s.t. \|\boldsymbol{a}\|_{l_p} < \delta_1, \|\boldsymbol{b}\|_{l_p} < \delta_2, \sum a_i = 1 \end{aligned} \tag{5.9}$$

5.3 Regularized hull based ISCRC

In Section 6.4.2, we introduced CRSSD, and presented two important instantiations of it, i.e., convex hull based CRSSD and regularized hull based CRSSD. With either one of them, the ISCRC (refer to Eq. (5.5)) can be implemented to perform ISFR. In this section, we discuss the minimization of regularized hull based CRSSD model, and the corresponding classification scheme is called regularized hull based ISCRC, denoted by RH-ISCRC. The minimization of convex hull based CRSSD and the corresponding classification scheme will be discussed in Section **??**.

5.3.1 Main model

We can re-write the regularized hull based CRSSD model in Eq. (5.9) as its Lagrangian formulation:

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{b}} \|\boldsymbol{Y}\boldsymbol{a} - \boldsymbol{X}\boldsymbol{b}\|_2^2 + \lambda_1\|\boldsymbol{a}\|_{l_p} + \lambda_2\|\boldsymbol{b}\|_{l_p} \\ &s.t. \sum a_i = 1 \end{aligned} \tag{5.10}$$

where λ_1 and λ_2 are positive constants to balance the representation residual and the regularizer.

In ISFR, each gallery set $\boldsymbol{X}_k$ often has tens to hundreds of sample images so that the whole set $\boldsymbol{X}$ can be very big, making the computational cost to solve Eq. (5.10) very high. Considering the fact that the images in each set $\boldsymbol{X}_k$ have high redundancy, we can compress $\boldsymbol{X}_k$ into a much more compact set, denoted by $\boldsymbol{D}_k$, via dictionary learning methods [141], such as KSVD [157] and metaface learning [218]. Let $\boldsymbol{D} = [\boldsymbol{D}_1, ..., \boldsymbol{D}_k, ..., \boldsymbol{D}_K]$. We can then replace $\boldsymbol{X}$ by $\boldsymbol{D}$ in Eq. (5.10) to compute the regularized hull based CRSSD:

$$\begin{aligned} &(\hat{\boldsymbol{a}}, \hat{\boldsymbol{\beta}}) = \arg\min_{\boldsymbol{a},\boldsymbol{\beta}} \left\{ \begin{array}{l} \|\boldsymbol{Y}\boldsymbol{a} - \boldsymbol{D}\boldsymbol{\beta}\|_2^2 + \\ \lambda_1\|\boldsymbol{a}\|_{l_p} + \lambda_2\|\boldsymbol{\beta}\|_{l_p} \end{array} \right\} \\ &s.t. \sum a_i = 1 \end{aligned} \tag{5.11}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1; ...; \boldsymbol{\beta}_k; ...; \boldsymbol{\beta}_K]$ and $\boldsymbol{\beta}_k$ is the sub-vector of coefficients associated with $\boldsymbol{D}_k$. Based on our experimental results, compressing $\boldsymbol{X}_k$ into $\boldsymbol{D}_k$ significantly improve the speed with almost the same ISFR rate.

Either l_1-norm or l_2-norm can be used to regularize $\boldsymbol{a}$ and $\boldsymbol{\beta}$, while l_1-regularization will lead to sparser solutions but with more computational cost. Like in l_1-SVM [230] and SRC [201], sparsity can enhance the classification rate if the features are

not informative enough. Note that if the query set $\boldsymbol{Y}$ has only one sample, then $\boldsymbol{a} = 1$ and the proposed model in Eq. (5.11) will be reduced to the SRC (for l_1-regularization) or CRC (for l_2-regularization) scheme. Next, we present the optimization of l_2-norm and l_1-norm regularized hull based ISCRC in Section 5.3.2 and Section 5.3.3, respectively.

5.3.2 l_2-norm regularized hull based ISCRC

When l_2-norm is used to regularize $\boldsymbol{a}$ and $\boldsymbol{\beta}$, the problem in Eq. (5.11) has a closed-form solution. The Lagrangian function of Eq. (5.11) becomes

$$\begin{aligned} L(\boldsymbol{a},\boldsymbol{\beta},\lambda_3) &= \|\boldsymbol{Y}\boldsymbol{a} - \boldsymbol{D}\boldsymbol{\beta}\|_2^2 + \lambda_1 \|\boldsymbol{a}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_2^2 \\ &+\lambda_3(\boldsymbol{e}\boldsymbol{a} - 1) \\ &= \left\| [\boldsymbol{Y} \;\; -\boldsymbol{D}] \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{\beta} \end{bmatrix} \right\|_2^2 + \begin{bmatrix} \boldsymbol{a}^T & \boldsymbol{\beta}^T \end{bmatrix} \begin{bmatrix} \lambda_1 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \lambda_2 \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{\beta} \end{bmatrix} \\ &+\lambda_3([\boldsymbol{e} \;\; \boldsymbol{0}] \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{\beta} \end{bmatrix} - 1) \end{aligned} \tag{5.12}$$

where $\boldsymbol{e}$ is a row vector whose elements are 1.

Let $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{\beta} \end{bmatrix}$, $\boldsymbol{A} = [\boldsymbol{Y} \;\; -\boldsymbol{D}]$, $\boldsymbol{B} = \begin{bmatrix} \lambda_1 \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \lambda_2 \boldsymbol{I} \end{bmatrix}$ and $\boldsymbol{d} = [\boldsymbol{e} \;\; \boldsymbol{0}]^T$. Then Eq. (5.12) becomes:

$$L(\boldsymbol{z},\lambda_3) = \boldsymbol{z}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{B}\boldsymbol{z} + \lambda_3(\boldsymbol{d}^T\boldsymbol{z} - 1) \tag{5.13}$$

There are

$$\frac{\partial L}{\partial \lambda_3} = \boldsymbol{d}^T\boldsymbol{z} - 1 = 0 \tag{5.14}$$

$$\frac{\partial L}{\partial z} = \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{z} + \lambda_3\boldsymbol{d} = 0 \tag{5.15}$$

According to Eq. (5.14) and Eq. (5.15), we get the closed form solution to Eq. (5.12):

$$\hat{\boldsymbol{z}} = \begin{bmatrix} \hat{\boldsymbol{a}} \\ \hat{\boldsymbol{\beta}} \end{bmatrix} = \boldsymbol{z}_0 / \boldsymbol{d}^T\boldsymbol{z}_0 \tag{5.16}$$

where $\boldsymbol{z}_0 = (\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B})^{-1}\boldsymbol{d}$.

After $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{\beta}}$ are got, the distance between query set $\boldsymbol{Y}$ and a gallery set $\boldsymbol{X}_k$ is calculated as $r_k = \left\|\boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k\right\|_2^2$, and then the class label of $\boldsymbol{Y}$ is determined by Eq. (5.5). For RH-ISCRC-l_2, the main time consumption is to solve the inverse of matrix $(\boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{B})$. Hence, the time complexity of RH-ISCRC-l_2 is $O(\left(n_a + n_\beta\right)^3)$, where n_a is the number of sample images in $\boldsymbol{Y}$ and n_β is the number of atoms in $\boldsymbol{D}$.

The CRNP method [203] also collaboratively represents the query set over the gallery sets. The differences between the proposed RH-ISCRC-l_2 and CRNP lie in the optimization procedure and the classification rule. RH-ISCRC-l_2 has a closed-form solution while CRNP adopts the same optimization method as RNP [220], which iteratively converges to the global optimal solution. Besides, CRNP uses the same classification rule as RNP, which utilizes both the reconstruction error and rank of image set matrix. RH-ISCRC-l_2 only uses the reconstruction error for classification.

5.3.3 l_1-norm regularized hull based ISCRC

When l_1-norm regularization is used, we use the alternating minimization method, which is very efficient to solve multiple variable optimization problems [68]. For

Eq. (5.11), we have the following augmented Lagrangian function:

$$\begin{aligned} L(\boldsymbol{a},\boldsymbol{\beta},\lambda) &= \|\boldsymbol{Y}\boldsymbol{a}-\boldsymbol{D}\boldsymbol{\beta}\|_2^2+\lambda_1\|\boldsymbol{a}\|_1+\lambda_2\|\boldsymbol{\beta}\|_1 \\ &+<\lambda,\boldsymbol{e}\boldsymbol{a}-1>+\tfrac{\gamma}{2}\|\boldsymbol{e}\boldsymbol{a}-1\|_2^2 \end{aligned} \tag{5.17}$$

where λ is the Lagrange multiplier, $\langle\cdot,\cdot\rangle$ is the inner product, and $\gamma > 0$ is the penalty parameter.

Then $\boldsymbol{a}$ and $\boldsymbol{\beta}$ are optimized alternatively with the other one fixed. More specifically, the iterations of minimizing $\boldsymbol{a}$ go as follows:

$$\begin{aligned} \boldsymbol{a}^{(t+1)} &= \arg\min_{\boldsymbol{a}} L(\boldsymbol{a},\boldsymbol{\beta}^{(t)},\lambda^{(t)}) \\ &= \arg\min_{\boldsymbol{a}} f(\boldsymbol{a})+\tfrac{\gamma}{2}\left\|\boldsymbol{e}\boldsymbol{a}-1+\lambda^{(t)}/\gamma\right\|_2^2 \\ &= \arg\min_{\boldsymbol{a}} \left\|\tilde{\boldsymbol{Y}}\boldsymbol{a}-\boldsymbol{x}\right\|_2^2+\lambda_1\|\boldsymbol{a}\|_1 \end{aligned} \tag{5.18}$$

where $f(\boldsymbol{a}) = \left\|\boldsymbol{Y}\boldsymbol{a}-\boldsymbol{D}\boldsymbol{\beta}^{(t)}\right\|_2^2+\lambda_1\|\boldsymbol{a}\|_{l_p}$, $\tilde{\boldsymbol{Y}} = \left[\boldsymbol{Y};(\gamma/2)^{1/2}\boldsymbol{e}\right]$, $\boldsymbol{x} = [\boldsymbol{D}\boldsymbol{\beta}^{(t)};(\gamma/2)^{1/2}(1-\lambda^{(t)}/\gamma)]$.

The problem in Eq. (5.18) can be easily solved by some representative l_1-minimization approaches such as LARS [45].

After $\boldsymbol{a}^{(t+1)}$ is updated, $\boldsymbol{\beta}^{(t+1)}$ can be obtained by solving another l_1-regularized optimization problem:

$$\begin{aligned} \boldsymbol{\beta}^{(t+1)} &= \arg\min_{\boldsymbol{\beta}} L(\boldsymbol{a}^{(t+1)},\boldsymbol{\beta},\lambda^{t}) \\ &= \arg\min_{\boldsymbol{\beta}} \left\|\boldsymbol{Y}\boldsymbol{a}^{(t+1)}-\boldsymbol{D}\boldsymbol{\beta}\right\|_2^2+\lambda_2\|\boldsymbol{\beta}\|_1 \end{aligned} \tag{5.19}$$

Once $\boldsymbol{a}^{(t+1)}$ and $\boldsymbol{\beta}^{(t+1)}$ are got, λ is updated as follows:

$$\lambda^{(t+1)} = \lambda^{(t)}+\gamma\left(\boldsymbol{e}\boldsymbol{a}^{(t+1)}-1\right) \tag{5.20}$$

The algorithm of RH-ISCRC-l_1 for ISFR is summarized in Table 5.2 and it converges. The problem in Eq. (5.17) is convex, and the subproblems in Eq. (5.18)

and Eq. (5.19) are convex and can be solved using the LARS algorithm. It had been shown in [133], for the general convex problem, the alternating minimization approach would converge to the correct solution. One curve of the objective function value of RH-ISCRC-l_1 versus the iteration number is shown in Fig. 5.6. Honda/USCD database [105] is also used. The query set $\boldsymbol{Y}$ and each gallery set $\boldsymbol{X}_k$ has 200 frames. Note that one image set is acquired from one video clip and there is no intersection between the query set and each gallery set. We compress each set $\boldsymbol{X}_k$ into a dictionary $\boldsymbol{D}_k$ with 20 atoms by using the metaface learning method [218]. Since there are 20 gallery sets, the set $\boldsymbol{D} = [\boldsymbol{D}_1, ..., \boldsymbol{D}_k, ..., \boldsymbol{D}_{20}]$ has 20×20=400 atoms. From the figure we can see that RH-ISCRC-l_1 converges after about five iterations.

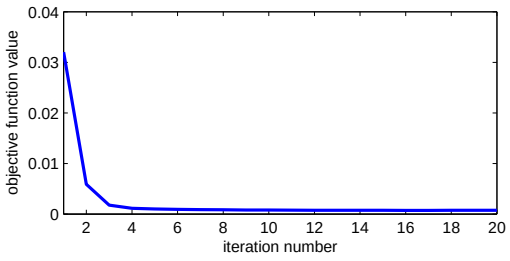


Figure 5.6 Convergence of RH-ISCRC-l_1.

Since the complexity of sparse coding is $O(m^2 n^{\varepsilon})$, where m is the feature dimension, n is the atom number and $\varepsilon \geq 1.2$ [96], we can get that the time complexity of RH-ISCRC-l_1 is $O(lm^2({n_a}^{\varepsilon} + {n_\beta}^{\varepsilon}))$, where n_a is the number of samples in $\boldsymbol{Y}$, n_β is the number of atoms in $\boldsymbol{D}$ and l is the iteration number.

Table 5.2 Algorithm of RH-ISCRC for ISFR.

Input: query set $\boldsymbol{Y}$; gallery sets $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_k, ..., \boldsymbol{X}_K]$, λ_1 and λ_2.

Output: the label of query set $\boldsymbol{Y}$.

Initialize $\beta^{(0)}$, $\lambda^{(0)}$ and $0 \leftarrow t$.

Compress $\boldsymbol{X}_k$ to $\boldsymbol{D}_k$, $k = 1, 2, ..., K$ using metaface learning [218].

While $t < max\_num$ do

Step 1: Update $\boldsymbol{a}$ by Eq. (5.18);

Step 2: Update $\boldsymbol{\beta}$ by Eq. (5.19);

Step 3: Update λ by Eq. (5.20);

Step 4: $t \leftarrow t + 1$.

End while

Compute $r_k = \left\| \boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k \right\|_2^2$, $k = 1, 2, ...K$.

Identity($\boldsymbol{Y}$)=arg $\min_k\{r_k\}$.

5.3.4 Examples and discussions

Let's use an example to better illustrate the classification process of RH-ISCRC. We use the Honda/USCD database [105]. The experiment setting is the same as Fig. 5.6. By Eq. (5.11), the computed coefficients in $\boldsymbol{a}$ and $\boldsymbol{\beta}$ are plotted in Fig. 5.7 (by l_1-regularization) and Fig. 5.8 (by l_2-regularization), respectively. The highlighted coefficients in the figures are associated with set $\boldsymbol{X}_{10}$, which has the same class label as $\boldsymbol{Y}$. Clearly, these coefficients are much more significant than the coefficients associated with the other classes. Meanwhile, from Fig. 5.7 and Fig. 5.8 we can see that l_1-regularized hull based CRSSD leads to sparser $\boldsymbol{a}$ and $\boldsymbol{\beta}$, implying that

only few samples are dominantly involved in representation and classification.

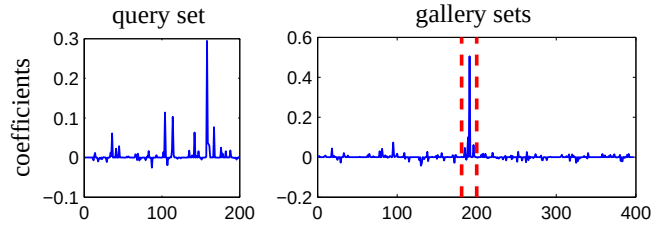


Figure 5.7 The coefficient vectors $\hat{\boldsymbol{a}}$ (of $\boldsymbol{Y}$) and $\hat{\boldsymbol{\beta}}$ (of $\boldsymbol{D}$) by l_1-regularized hull based CRSSD.

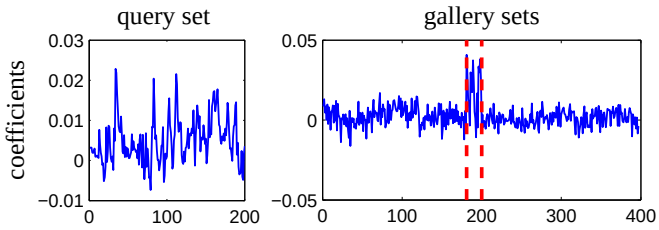


Figure 5.8 The coefficient vectors $\hat{\boldsymbol{a}}$ (of $\boldsymbol{Y}$) and $\hat{\boldsymbol{\beta}}$ (of $\boldsymbol{D}$) by l_2-regularized hull based CRSSD.

In Fig. 5.9, we show the reconstructed faces by $\boldsymbol{Y}\hat{\boldsymbol{a}}$ with l_1-regularized hull based CRSSD. The distances between $\boldsymbol{Y}\hat{\boldsymbol{a}}$ and each $\boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k$, i.e., r_k, are also given. We see that r_{10} is 0.03, which is the minimal one among all the gallery sets, meaning that ISCRC will make the correct recognition. Here the relationships between ISCRC and manifold based methods can be revealed. MMD assumes that an image set can be modeled as a set of local subspaces so that the image set distance is defined as the weighted average distance between any two local subspaces [193]. The distance between two local subspaces is related to the cluster exemplar and principle angel. Correspondingly, ISCRC seeks for a local subspace ($\boldsymbol{Y}\hat{\boldsymbol{a}}$) in the query image set and a local subspace ($\boldsymbol{D}\hat{\boldsymbol{\beta}}$) in all the gallery sets, as shown in Fig. 5.7 .

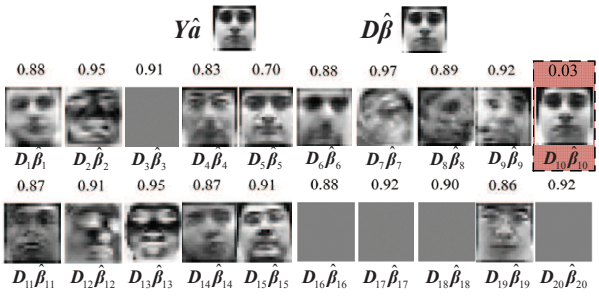


Figure 5.9 Reconstructed faces $\boldsymbol{Y}\hat{\boldsymbol{a}}$, $\boldsymbol{D}\hat{\boldsymbol{\beta}}$, $\boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k$ (we normalized each $\boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k$ for better visualization). The number over each $\boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k$ is the residual $r_k = \left\|\boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k\right\|_2^2$.

In classification, the distance between the query set and the template set of the k^{th} class is the distance between the local subspace ($\boldsymbol{Y}\hat{\boldsymbol{a}}$) and the local subspace $\boldsymbol{D}_k\hat{\boldsymbol{\beta}}_k$.

5.4 Kernelized convex hull based ISCRC

We then focus on how to compute the convex hull based CRSSD in Eq. (5.6) and use it for ISCRC. Since there can be many sample images in gallery sets, $\boldsymbol{X}$ can be a fat matrix (note that usually we use a low dimensional feature vector to represent each face image). Even we compress $\boldsymbol{X}$ into a more compact set $\boldsymbol{D}$, the system can still be under-determined. In Section 3 we imposed the l_p-norm regularization on $\boldsymbol{a}$ and $\boldsymbol{b}$ to make the solution stable. When the convex hull is used, however, the constraint may not be strong enough to get a stable solution of Eq. (5.6). In addition, if the underlying relationship between the query set and gallery sets is highly nonlinear, it is difficult to approximate the hull of query set as a linear combination of gallery sets.

One simple solution to solving both the above two problems is the kernel trick;

that is, we can map the data into a higher dimensional space where the subjects can be approximately linearly separable. The mapped gallery data matrix in the high-dimensional space will be generally over-determined. In such a case, the convex hull constraint will be strong enough for a stable solution. The kernelized convex hull based CRSSD model is:

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{\beta}} \left\|\phi(\boldsymbol{Y})\boldsymbol{a} - [\phi(\boldsymbol{D}_1), \phi(\boldsymbol{D}_2), ..., \phi(\boldsymbol{D}_K)]\,\boldsymbol{\beta}\right\|^2 \\ &s.t.\ \textstyle\sum a_i = 1, \sum \beta_j = 1, \\ &\quad 0 \le a_i \le \tau, i = 1, ..., n_a, \\ &\quad 0 \le \beta_j \le \tau, j = 1, ..., n_\beta. \end{aligned} \tag{5.21}$$

The above minimization can be easily solved by the standard quadratic optimization (QP [35]) method. The solution exhibits global and quadratic convergence, as proved in [35]. Different kernel functions can be used, e.g., linear kernel and Gaussian kernel. We call the corresponding method kernelized convex hull based ISCRC, denoted by KCH-ISCRC. The classification rule is the same as RH-ISCRC with $r_k = \left\|\phi(\boldsymbol{Y})\hat{\boldsymbol{a}} - \phi(\boldsymbol{D}_k)\hat{\boldsymbol{\beta}}_k\right\|_2^2$. As convex hull based CRSSD is to solve a convex QP problem, the time complexity of KCH-ISCRC is $O((n_\beta + n_a)^3)$, which is similar to SVM. The algorithm of KCH-ISCRC is given in Table 5.3. To reduce the computational cost, the kernel matrix $k(\boldsymbol{D}, \boldsymbol{D})$ can be computed and stored. When a query set $\boldsymbol{Y}$ comes, we only need to calculate $k(\boldsymbol{Y}, \boldsymbol{Y})$ and $k(\boldsymbol{Y}, \boldsymbol{D})$.

Like in Fig. 5.7 and Fig. 5.8, in Fig. 5.10 we show the coefficient vectors $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{\beta}}$ solved by Eq. (5.21). The Gaussian kernel is used and the experimental setting is the same as that in Figs. 5.7 and 5.8 (the only difference is that each compressed gallery set $\boldsymbol{D}_k$ has 50 atoms). We can see that the coefficients associated with gallery set $\boldsymbol{D}_{10}$ are larger than the other gallery sets, resulting in a smaller representation

Table 5.3 Algorithm of KCH-ISCRC for ISFR.

Input: query set $\boldsymbol{Y}$; gallery sets $\boldsymbol{X} = [\boldsymbol{X}_1, ..., \boldsymbol{X}_k, ..., \boldsymbol{X}_K]$, τ.
Output: the label of query set $\boldsymbol{Y}$.
Compress $\boldsymbol{X}_k$ to $\boldsymbol{D}_k$, $k = 1, 2, ..., K$ by metaface learning [220];
Solve the QP problem in Eq. (5.21);
Compute $r_k = \left\|\phi(\boldsymbol{Y})\hat{\boldsymbol{a}} - \phi(\boldsymbol{D}_k)\hat{\boldsymbol{\beta}}_k\right\|_2^2$, $k = 1, 2, ...K$;
Identity($\boldsymbol{Y}$)=arg $\min_k\{r_k\}$.

residual and hence the correct recognition.

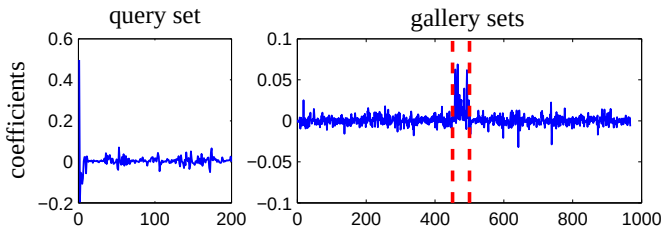


Figure 5.10 The coefficient vectors $\hat{\boldsymbol{a}}$ (of $\boldsymbol{Y}$) and $\hat{\boldsymbol{\beta}}$ (of $\boldsymbol{D}$) by kernelized convex hull based CRSSD.

5.5 Experimental analysis

We used the Honda/UCSD [105], CMU Mobo [65], and Youtube Celebrities [95] datasets to test the performance of the proposed method. The comparison methods fall into four categories:

C1. Subspace and manifold based methods: Mutual Subspace Method (MSM)

[206], Discriminant Canonical Correlations (DCC[2]) [98], Manifold-Manifold Distance (MMD[3]) [193], and Manifold Discriminant Analysis (MDA[4]) [190].

C2. Affine/convex hull based methods: Affine Hull based Image Set Distance (AHISD[5]) [21], Convex Hull based Image Set Distance (CHISD[6]) [21], Sparse Approximated Nearest Points (SANP[7]) [78], and Regularized Nearest Points (RNP) [220].

C3. Representation based methods: Sparse Representation based Classifier (SRC) [201], Collaborative Representation based Classifier (CRC) [226]. We tested to use the average and minimal representation residual of query set for classification and found that average residual works better. Hence in this chapter, the average residual is used in SRC/CRC for classification.

C4. Kernel methods: KSRC (Kernel SRC) [55], KCRC (Kernel CRC) [216], AHISD [21], and CHISD [21]. For KSRC and KCRC, the average residual is used for classification.

For the proposed methods, RH-ISCRC is compared with those non-kernel methods and KCH-ISCRC is compared with those kernel methods.

[2]http://www.iis.ee.ic.ac.uk/ tkkim/code.htm
[3]http://www.jdl.ac.cn/user/rpwang/research.htm
[4]http://www.jdl.ac.cn/user/rpwang/research.htm
[5]http://www2.ogu.edu.tr/ mlcv/softwareimageset.html
[6]http://www2.ogu.edu.tr/ mlcv/softwareimageset.html
[7]https://sites.google.com/site/yiqunhu/cresearch/sanp

5.5.1 Parameter setting

For competing methods, the important parameters were empirically tuned according to the recommendations in the original literature for fair comparison. For DCC [98], if there is only one set per class, then the training set is divided into two sets since at least two sets per class are needed in DCC. For MMD, the number of local models is set following the work in [193]. For MDA, there are three parameters, i.e., the number of local models, the number of between-class NN local models and the subspace dimension. The three parameters are configured according to the work in [190]. For SANP, we adopted the same parameters as [78]. For SRC, CRC, KSRC and KCRC, λ that balances the residual and regularization is tuned from $[0.01, 0.001, 0.0001]$. For AHISD and CHISD, C is set as 100. For all kernel methods, Gaussian kernel ($k(x, y) = \exp(-\|x - y\|_2^2 / 2\delta^2)$) is used, and δ is set as 5. The experiments of 50 frames, 100 frames and 200 frames per set are conducted on the three databases. If the number of samples in the set is less than the given number, then all the samples in the set are used.

For the proposed RH-ISCRC, we set $\lambda_1 = 0.001$, $\lambda_2 = 0.001$, $\lambda = 2.5/n_a$ (n_a is the number of samples in the query set), $\gamma = \lambda/2$. The number of atoms in the compressed set $\boldsymbol{D}_k$ is set as 20 on Honda/UCSD and 10 on CMU MoBo and YouTube. For KCH-ISCRC, $\tau = 1$ and the number of atoms in each $\boldsymbol{D}_k$ is set as 50 for all datasets. The sensitivity of the proposed methods to parameters will be discussed in Section 5.5.6.

5.5.2 Honda/UCSD

The Honda/UCSD dataset consists of 59 video sequences involving 20 different subjects [105]. The Viola-Jones face detector [187] is used to detect the faces in each frame and resize the detected faces to 20×20 images. Some examples of Honda/UCSD dataset are shown in Figure 5.11. Histogram equalization is utilized to reduce the illumination variations. Our experiment setting is the same as [105][78]: 20 sequences are set aside for training and the remaining 39 sequences for testing. The intensity is used as the feature.

Figure 5.11 Some examples of Honda/UCSD dataset.

Table 5.4 Recognition rates on Honda/UCSD (%).

Non-kernel	50	100	200	Year
MSM [206]	74.36	79.49	89.74	1998
DCC [98]	76.92	84.62	94.87	2007
MMD [193]	69.23	87.18	94.87	2008
MDA [190]	82.05	94.87	97.44	2009
SRC [201]	84.62	92.31	92.31	2009
AHISD [21]	82.05	84.62	89.74	2010
CHISD [21]	82.05	84.62	92.31	2010
SANP [78]	84.62	92.31	94.87	2011
CRC [226]	84.62	94.87	94.87	2011
RNP [220]	87.18	94.87	**100.0**	2011
RH-ISCRC-l_1	**89.74**	**97.44**	**100.0**	
RH-ISCRC-l_2	**89.74**	**97.44**	**100.0**	
Kernel	50	100	200	Year
AHISD [21]	84.62	84.62	82.05	2010
CHISD [21]	84.62	87.18	89.74	2010
KSRC [55]	87.18	**97.44**	97.44	2009
KCRC [216]	82.05	94.87	94.87	2012
KCH-ISCRC	**89.74**	94.87	**100.0**	

The experimental results are listed in Table 5.4. We can see that for those non-kernel methods, the proposed RH-ISCRC outperforms much all the other methods. Note that in [21], kernel CHISD achieves 100% recognition accuracy when all the

frames in one video clip are used. In this chapter, following the experiment setting of SANP [78], we reported the accuracy using different number of frames per set. When 200 frames per set are used, both RH-ISCRC and KCH-ISCRC achieve 100% accuracy, which shows the superiority to CHISD and AHISD. For the kernel based method, the proposed KCH-ISCRC performs the best except for the case when 100 frames per set are used. We can also see that on this dataset, RH-ISCRC-l_1 and RH-ISCRC-l_2 achieve the same recognition rate, which implies that on this dataset the l_2-norm regularization is strong enough to yield a good solution to the regularized hull based CRSSD in Eq. (5.11).

5.5.3 CMU MoBo

The CMU Mobo[8] (Motion of Body) dataset [65] was originally established for human pose identification and it contains 96 sequences from 24 subjects. Four video sequences are collected per subject, each of which corresponds to a walking pattern. Again, the Viola-Jones face detector [187] is used to detect the faces and the detected face images are resized to 40×40. The LBP feature is used, which is the same as the work in [21] and [78].

One video sequence per subject is selected for training while the rest are used for testing. Ten-fold cross validation experiments are conducted and the average recognition results are shown in Table 5.5. We can clearly see that the proposed methods outperform the other methods under different frames per set. On this dataset and the Honda/UCSD dataset, the proposed non-kernel RH-ISCRC and the kernel based KCH-ISCRC have similar ISFR rates.

[8] http://www.ri.cmu.edu/publication_view.html?pub_id=3904

Table 5.5 Recognition rates on CMU MoBo(%).

Non-kernel	50	100	200	Year
MSM [206]	84.3 ± 2.6	86.6±2.2	89.9±2.4	1998
DCC [98]	82.1± 2.7	85.5±2.8	91.6±2.5	2007
MMD [193]	86.2 ±2.9	94.6±1.9	96.4±0.7	2008
MDA [190]	86.2 ±2.9	93.2±2.8	95.8±2.3	2009
SRC [201]	91.0 ±2.1	91.8±2.7	96.5±2.5	2009
AHISD [21]	91.6 ±2.8	94.1±2.0	91.9±2.6	2010
CHISD [21]	91.2 ±3.1	93.8±2.5	96.0±1.3	2010
SANP [78]	91.9 ±2.7	94.2±2.1	97.3±1.3	2011
CRC [226]	89.6 ±1.8	92.4±3.7	96.4±2.8	2011
RNP [220]	91.9 ±2.5	94.7±1.2	97.4±1.5	2013
RH-ISCRC-l_1	**93.5±2.8**	**96.5±1.9**	**98.7±1.7**	
RH-ISCRC-l_2	**93.5±2.8**	**96.4±1.9**	**98.4±1.7**	
Kernel	50	100	200	Year
AHISD [21]	88.9±1.7	92.4±2.8	93.5±4.2	2010
CHISD [21]	91.5±2.0	93.4±4.0	97.4±1.9	2010
KSRC [55]	91.6 ±2.8	94.1±2.0	96.8±2.0	2010
KCRC [216]	91.2 ±3.1	93.4±2.9	96.6±2.6	2012
KCH-ISCRC	**94.2 ±2.1**	**96.4±2.3**	**98.4±1.9**	

5.5.4 YouTube Celebrities

The YouTube Celebrities[9] is a large scale video dataset collected for face tracking and recognition, consisting of 1,910 video sequences of 47 celebrities from YouTube [95]. As the videos were captured in unconstrained environments, the recognition task becomes much more challenging due to the larger variations in pose, illumination and expressions. The face in each frame is also detected by the Viola-Jones face detector and resized to a 30×30 gray-scale image. The intensity value is used as feature. The experiment setting is the same as [78, 190, 191]. Three video sequences per subject are selected for training and six for testing. Five-fold cross validation experiments are conducted.

The experimental results are shown in Table 6.12. It can be seen that among the non-kernel methods, the proposed RH-ISCRC-l_1 achieves the highest recognition rate, while among the kernel based methods, the proposed KCH-ISCRC performs the best. Since this Youtube Celebrities dataset was established under uncontrolled environment, there are significant variations among the query and gallery sets, and therefore the l_1-regularization is very helpful to improve the stability and discrimination of the solution to Eq. (5.11). As a consequence, RH-ISCRC-l_1 leads to much better results than RH-ISCRC-l_2 on this dataset. On the other hand, the kernel based KCH-ISCRC leads to better results than RH-ISCRC in this experiment. Besides, the number of frames per set also affect the performance of ISCRC. When number of frames is small, the improvement by ISCRC is more significant.

[9] http://seqam.rutgers.edu/site/index.php?option=com_content&view=article&id=64&Itemid=80

Table 5.6 Recognition rates on YouTube (V1 %).

Non-kernel	50	100	200	Year
MSM [206]	54.8±8.7	57.4±7.7	56.7±6.9	1998
DCC [98]	57.6±8.0	62.7±6.8	65.7±7.0	2007
MMD [193]	57.8±6.6	62.8±6.2	64.7±6.3	2008
SRC [201]	61.5±6.9	64.4±6.8	66.0±6.7	2009
MDA [190]	58.5±6.2	63.3±6.1	65.4±6.6	2009
AHISD [21]	57.5±7.9	59.7±7.2	57.0±5.5	2010
CHISD [21]	58.0±8.2	62.8±8.1	64.8±7.1	2010
SANP [78]	57.8±7.2	63.1±8.0	65.6±7.9	2011
CRC [226]	56.5±7.4	59.5±6.6	61.4±6.4	2011
RNP [220]	59.9 ±7.3	63.3±8.1	64.4±7.8	2013
RH-ISCRC-l_1	**62.3±6.2**	**65.6±6.7**	**66.7±6.4**	
RH-ISCRC-l_2	57.4±7.2	60.7±6.5	61.4±6.4	
Kernel	50	100	200	Year
AHISD [21]	57.2±7.5	59.6±7.4	61.8±7.3	2010
CHISD [21]	57.9±8.3	62.6±8.1	64.9±7.2	2010
KSRC [55]	61.4±7.0	65.9±6.9	67.8±6.4	2010
KCRC [216]	57.5±7.9	60.6±6.8	62.7±7.7	2012
KCH-ISCRC	**64.5±7.6**	**67.4±8.0**	**69.7±7.4**	

5.5.5 Time comparison

Then let's compare the efficiency of competing methods. The Matlab codes of all competing methods are obtained from the original authors, and we run them on an Intel(R) Core(TM) i7-2600K (3.4GHz) PC. The average running time per set on CMU MoBo (200 frames per set) is listed in Table 6.11. We can see that the proposed RH-ISCRC-l_2 is the fastest among all competing methods except for RNP, while RH-ISCRC-l_1 also has a fast speed. Among all the kernel based methods, the proposed KCH-ISCRC is much faster than others. Overall, the proposed RH-ISCRC and KCH-ISCRC methods have not only high ISFR accuracy but also high efficiency than the competing methods.

5.5.6 Parameter sensitivity analysis

To verify if the proposed methods are sensitive to parameters, in this section we present the recognition accuracies with different parameter values. For RH-ISCRC, there are two parameters, λ_1 and λ_2 in Eq. (5.17), which need to be set. For KCH-ISCRC, there is only one parameter τ in Eq. (5.5). We show the recognition accuracies versus the parameters on the CMU MoBo dataset in Fig. 5.12, Fig. 5.13 and Fig. 5.14, respectively, for RH-ISCRC-l_1, RH-ISCRC-l_2 and KCH-ISCRC. The different colors correspond to different accuracies, as shown in the color bar. λ_1 and λ_2 are selected from $\{0.0005, 0.001, 0.01, 0.05\}$. In Fig. 5.12 and Fig. 5.13, the top sub-figure is for 50 frames per set, the middle is for 100 frames per set and the bottom corresponds to 200 frames per set. From Fig. 5.12, we can see that the accuracy of RH-ISCRC-l_1 is very stable when λ_1 varies from 0.0005 to 0.05

Table 5.7 Average running time per set on CMU MoBo (s).

Non-kernel	Time	Kernel	Time
MSM [206]	0.338	AHISD [21]	18.546
DCC [98]	0.349	CHISD [21]	18.166
MMD [193]	3.216	KSRC [55]	35.508
SRC [201]	5.301	KCRC [216]	6.543
MDA [190]	2.035	KCH-ISCRC	**2.03**
AHISD [21]	31.365		
CHISD [21]	18.029		
SANP [78]	11.124		
CRC [226]	0.684		
RNP [220]	**0.113**		
RH-ISCRC-l_1	0.788		
RH-ISCRC-l_2	**0.280**		

and λ_2 varies from 0.0005 to 0.01. When λ_2 is increased to 0.05, the recognition performance would degrade. Fig. 5.13 shows that RH-ISCRC-l_2 is insensitive to the values of λ_1 and λ_2. For example, in the experiments of 100 and 200 frames per set, the accuracy variation is within 0.5% for different λ_1 and λ_2. Considering the performance of both RH-ISCRC-l_1 and RH-ISCRC-l_2, λ_1 and λ_2 can both be set as 0.001. With this parameter setting, the accuracy is very stale in different experiments. For KCH-ISCRC, its recognition accuracies with different values of τ are shown in Fig. 5.14. τ is set as $\{1, 2, 5, 10, 50, 100\}$. One can see that KCH-ISCRC is insensitive to τ. Hence, we simply set τ as 1.

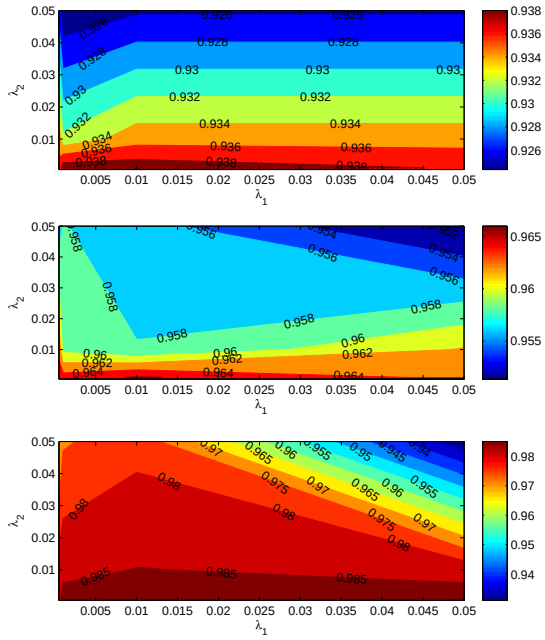


Figure 5.12 Recognition accuracy of RH-ISCRC-l_1 on CMU MoBo with different λ_1 and λ_2. Different colors represent different accuracy. Top: 50 frames per set; middle: 100 frames per set; bottom: 200 frames per set.

The dictionary learning technique is used in our method to compress each im-

age set to reduce the time complexity when representing a query image set. The number of atoms in the dictionary needs to be defined before dictionary learning. If the number of atoms is too small, the representation power of the dictionary will be reduced; if the number of atoms is large, the system tends to be under-determined and thus the solution may be less stable. We tested our algorithm by varying the number of atoms (for each sub-dictionary $\boldsymbol{D}_k$) from 5 to 50. The recognition accuracies versus the number of atoms on the CMU MoBo dataset are shown in Figs. 5.12-5.14. From Fig. 5.12 and Fig. 5.13, we can see that the recognition accuracies of both RH-ISCRC-l_1 and RH-ISCRC-l_2 vary little if the number of atoms is set within [10, 20]. From Fig. 5.14, we can see that for KCH-ISCRC the variation of recognition accuracies is within 0.5% under different number of atoms. This is because the feature dimension is relatively high in the kernel space and thus the solution is affected little by the dictionary size. Based on the above observation, in all our experiments we set the number of atoms as 10 or 20 for RH-ISCRC-l_1 and RH-ISCRC-l_2, and 50 for KCH-ISCRC.

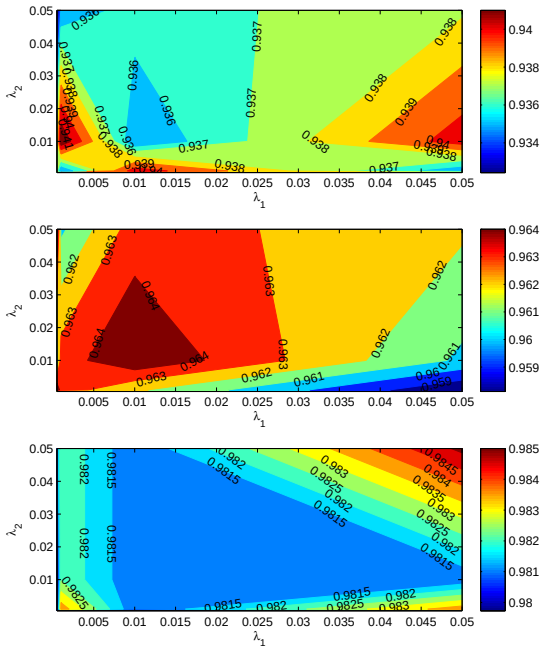


Figure 5.13 Recognition accuracy of RH-ISCRC-l_2 on CMU MoBo with different λ_1 and λ_2. Different colors represent different accuracy. Top: 50 frames per set; middle: 100 frames per set; bottom: 200 frames per set.

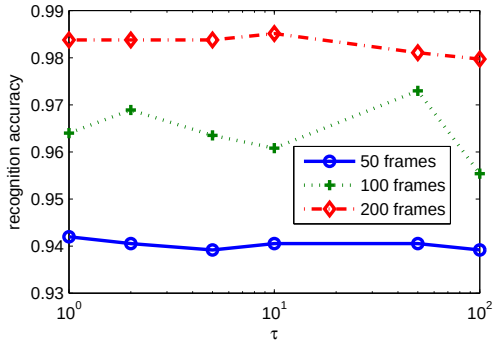


Figure 5.14 Recognition accuracy of KCH-ISCRC on CMU MoBo with different τ.

5.6 Conclusions and future work

We proposed a novel image set based collaborative representation and classification (ISCRC) scheme for image set based face recognition (ISFR). The query set was modeled as a convex or regularized hull, and a collaborative representation based set to sets distance (CRSSD) was defined by representing the hull of query set over all the gallery sets. The CRSSD considers the correlation and distinction of sample images within the query set and the relationship between the gallery sets. With CRSSD, the representation residual of the hull of query set by each gallery set can be computed and used for classification. Experiments on the three benchmark ISFR databases showed that the proposed ISCRC is superior to state-of-the-art ISFR methods in terms of both recognition rates and efficiency.

In this chapter, we proposed ISCRC to deal with video based face recognition tasks. Hulls are used to represent both the gallery face image sets and query face image set. However, for other image set classification tasks, e.g., multi-view object recognition, hull based representation may not be suitable. Hence, to extend the application of ISCRC, the representation of image sets should be modeled according to different tasks.

Chapter 6

From Point to Set: Extend the Learning of Distance Metrics

From Chapter 2 to Chapter 5, we have proposed patch based collaborative representation, local generic representation, regularized self-representation, and image set based collaborative representation models to solve small sample size problems, big sample size problems, and image set classification problems. The representation process of all these representation based classifiers is unsupervised and does not utilize the training label information. Actually, representation based classifiers, e.g., nearest subspace classifier, can be considered as a kind of point to set distance based classifiers. In this chapter, we propose to learn a discriminative point to set/set to set distance, which can enhance the performance of representation based classifiers.

6.1 Introduction

How to select a proper distance metric is a key problem in pattern classification, while the optimal distance metric for a specific pattern classification task depends on the underlying data structure and distributions. In recent years, it has been increasingly popular to learn a desired distance metric from the given training samples in many visual classification tasks, such as face/action/kinship verification [66], visual tracking [89], and image retrieval [1]. Metric learning methods can be categorized into unsupervised [33], semi-supervised [1] and supervised ones [66, 89], according to the availability of the class labels of training samples.

In general, metric learning aims to learn a valid distance metric, measured by which the samples from the positive sample pair (i.e., samples with the same class label or similar samples) could be as close as possible, while the samples from the negative sample pair (i.e., samples with the different class labels or dissimilar samples) could be as far as possible. Positive/negative sample pairs can be generated from the K nearest neighbors as in Large Margin Nearest Neighbor (LMNN) [196], Neighborhood Components Analysis (NCA) [63], or from the given sample pairs in verification as in Logistic Discriminative Metric Learning (LDML) [66], or from side information with some prior knowledge as in Information Theoretic Metric Learning (ITML) [41]. In some cases, only positive pairs are used in metric learning [125]. In [188], metric learning is formulated as a kernel classification model and the relations with LMNN and ITML are discussed. Metric learning algorithms have also been developed for multi-task learning [140], multiple instance learning [67] and nonlinear metrics [94].

Currently, almost all the metric learning methods focus on the learning of a

point-to-point distance (PPD) metric in couple with the nearest neighbor classifier (NNC). In many computer vision tasks (e.g., face recognition), however, we need to measure the distance between an image (i.e., a point) and an image set (i.e., a point set). In video based recognition tasks [193] or multi-view object recognition [98], we even need to measure the distance between two image sets. Therefore, it is highly desired to design effective point-to-set distance (PSD) and set-to-set distance (SSD) metric learning methods. Unfortunately, many PPD metric learning methods cannot be readily applied to PSD and SSD based classification.

A set is often modeled as a hull, a convex hull (CH), or an affine hull (AH), and PSD can then be defined as the distance from a point to this hull. Correspondingly, the nearest subspace classifier (NSC), nearest convex hull classifier (NCH) [186], and nearest convex affine classifier (NAH) [186] are proposed for PSD based classification. In [22], a set is modeled as a bounding hyperdisk (the set formed by intersecting their affine hull and their smallest bounding hypersphere), and a nearest hyperdisk classifier (NHD) is proposed for classification. Given a query sample, those PSD based classifiers (NSC, NCH, NAH and NHD) compute its distance to each class, i.e., the PSD between the query samples and the set of templates of this class, and classify it to the class with the minimal point-to-set distance.

The calculation of SSD also depends on the means to model a set. In [21], by modeling each set as a CH/AH, the CH/AH based image set distance (CHISD/AHISD) is defined. In [78], sparsity is imposed on the AH model and a sparse approximation nearest points (SANP) method is proposed for image set classification. In [220], a regularized affine hull (RAH) is proposed to model a set, and the SSD is defined between two RAHs. In [206], each set is represented by a linear

subspace and the angles between two subspaces are utilized to measure the similarity of two sets. The method in [98] employs canonical correlation to measure the similarity between two sets. In [193], an image set is modeled as a manifold and a manifold-to-manifold distance (MMD) is proposed. After calculating the distance from the query set to each template set, those SSD based classifiers classify the query set to the class with the minimal set-to-set distance. To introduce discriminative information to SSD, projection matrix is learned in a large margin manner, e.g., discriminative canonical correlation (DCC) [98] and manifold discriminant analysis (MDA) [190]. In [204], a set based discriminative ranking model is proposed by iterating between SSD finding and discriminative feature space projection.

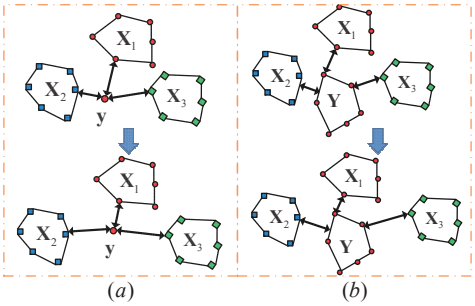


Figure 6.1 PSD (left) and SSD (right) Metric learning.

Despite that metric learning has been successfully used in PPD based classification, few attentions have been paid to PSD and SSD based classification. As shown in the upper part of Fig. 6.1(a), the query image $\boldsymbol{y}$ (represented as a red dot) has the same class label as template set $\boldsymbol{X}_1$ (represented as a red hull) but it will be misclassified since it has a closer PSD to set $\boldsymbol{X}_2$. If a proper metric learning method can be developed, it is possible that with the new distance metric, the PSD between $\boldsymbol{y}$ and $\boldsymbol{X}_1$ is smaller than that between $\boldsymbol{y}$ and $\boldsymbol{X}_2$, and consequently $\boldsymbol{y}$ can be correctly

classified, as shown in the bottom part of Fig. 6.1(a). Similar anticipation goes to the metric learning of SSD based classification, as illustrated in Fig. 6.1(b), where the query set $\boldsymbol{Y}$ can be correctly classified with some proper SSD based distance metric.

With the above considerations, in this chapter we propose two novel metric learning models, PSD metric learning (PSDML) and SSD metric learning (SSDML), to enhance the performance of PSD and SSD based classification. One image (or image set) and one similarly labeled image set construct a positive pair, while one image (or image set) and one differently labeled set construct a negative pair. Then the PSDML and SSDML problems are formulated as a sample pair classification problem. Each sample pair is characterized by the covariance matrix of its two samples, and a covariance kernel is introduced. A discriminative function is then proposed for sample pair classification, and finally the PSDML and SSDML can be solved by using an SVM model. The proposed PSDML and SSDML methods can effectively improve the performance of PSD and SSD based classification, and are much more efficient than state-of-the-art metric learning methods.

The main abbreviations used in this chapter are summarized in the following Table 6.1.

6.2 Set based distances

Before distance metric learning, we need to first define how the distance is measured. In this section, we describe how an image set is modeled, and how the corresponding point-to-set and set-to-set distances are defined.

Table 6.1 The main abbreviations used in this chapter.

PPD	point to point distance
PSD	point to set distance
SSD	set to set distance
PSDML	point to set distance metric learning
SSDML	set to set distance metric learning

6.2.1 Image set model

An image set is usually represented by a hull, i.e., a subspace spanned by all the available samples in the set. The hull of a set of samples $\boldsymbol{D} = [\boldsymbol{d}_1..., \boldsymbol{d}_i..., \boldsymbol{d}_n]$ is defined as $H(\boldsymbol{D}) = \{\boldsymbol{D}\boldsymbol{a}\}$, where $\boldsymbol{a} = [a_1, ..., a_i, ..., a_n]$. Usually, $\sum a_i = 1$ is required and a_i is required to be bounded:

$$H(\boldsymbol{D}) = \{\sum \boldsymbol{d}_i a_i | \sum a_i = 1, -\tau_1 \leq a_i \leq \tau_2\} \tag{6.1}$$

If $\tau_1 = -inf$ and $\tau_2 = inf$, $H(\boldsymbol{D})$ is an affine hull [186]. If $\tau_1 < 0$ and $\tau_2 > 0$, $H(\boldsymbol{D})$ is a reduced affine hull [21]. If $\tau_1 = 0$ and $\tau_2 = 1$, $H(\boldsymbol{D})$ is a convex hull [186]. If $\tau_1 = 0$ and $\tau_2 < 1$, $H(\boldsymbol{D})$ is a reduced convex hull [21].

To rule out the meaningless points which are too far from the sample mean, the regularized affine hull (RAH) [220] is defined as follows to model an image set:

$$H(\boldsymbol{D}) = \left\{\sum \boldsymbol{d}_i a_i | \sum a_i = 1, \|\boldsymbol{a}\|_{l_p} \leq \sigma\right\} \tag{6.2}$$

6.2.2 Point-to-set distance (PSD)

Given a sample $\boldsymbol{x}$ and a set of samples $\boldsymbol{D}$, a point to set distance $d(\boldsymbol{x}, \boldsymbol{D})$ between $\boldsymbol{x}$ and $\boldsymbol{D}$ can be defined as follows:

$$d(\boldsymbol{x}, \boldsymbol{D}) = \|\boldsymbol{x} - \boldsymbol{D}\hat{\boldsymbol{a}}\|_2 \tag{6.3}$$

where $\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \|\boldsymbol{x} - H(\boldsymbol{D})\|_2^2$. When $H(\boldsymbol{D})$ is a hull, the solution of $\min_{\boldsymbol{a}} \|\boldsymbol{x} - H(\boldsymbol{D})\|_2^2$ can be easily obtained by least square regression as $\left(\boldsymbol{D}^T\boldsymbol{D}\right)^{-1}\boldsymbol{D}^T\boldsymbol{x}$ if $\boldsymbol{D}^T\boldsymbol{D}$ is non-singular, or by ridge regression $\left(\boldsymbol{D}^T\boldsymbol{D} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{D}^T\boldsymbol{x}$ if $\boldsymbol{D}^T\boldsymbol{D}$ is (nearly) singular.

To make the PSD more accurate for classification, a projection matrix $\boldsymbol{P}$ can be introduced to project the samples into a desired space. The corresponding PSD distance, denoted by $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{D})$, is then defined as:

$$\begin{aligned} d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{D}) &= \|\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{D}\hat{\boldsymbol{a}})\|_2^2 \\ &= (\boldsymbol{x} - \boldsymbol{D}\hat{\boldsymbol{a}})^T\boldsymbol{P}^T\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{D}\hat{\boldsymbol{a}}) \\ &= (\boldsymbol{x} - \boldsymbol{D}\hat{\boldsymbol{a}})^T\boldsymbol{M}(\boldsymbol{x} - \boldsymbol{D}\hat{\boldsymbol{a}}) \end{aligned} \tag{6.4}$$

where $\hat{\boldsymbol{a}} = \arg\min_{\boldsymbol{a}} \|\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{D}\boldsymbol{a})\|_2^2$, and

$$\boldsymbol{M} = \boldsymbol{P}^T\boldsymbol{P}, \tag{6.5}$$

When $\hat{\boldsymbol{a}}$ is obtained, we can form a sample pair $(\boldsymbol{x}, \boldsymbol{D}\hat{\boldsymbol{a}})$. Clearly, the PSD $d_{\boldsymbol{M}}(\boldsymbol{x}, \boldsymbol{D})$ defined in Eq. (6.4) can be viewed as a Mahalanobis distance [41] between $\boldsymbol{x}$ and $\boldsymbol{D}\hat{\boldsymbol{a}}$, and the matrix $\boldsymbol{M}$ is always semi-positive definite.

In PSD based classification, the distance between the query sample $\boldsymbol{y}$ and the template set of each class $\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_c$ (c is the number of classes) needs to be computed first. Suppose that the nearest subspace classifier (NSC) is used. Given

$\boldsymbol{M}$, for class i, we have $\hat{\boldsymbol{a}}_i = \boldsymbol{W}_i\boldsymbol{y}$, where

$$\boldsymbol{W}_i = \left(\boldsymbol{X}_i^T\boldsymbol{M}\boldsymbol{X}_i + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}_i^T\boldsymbol{M}. \tag{6.6}$$

and then the PSD between $\boldsymbol{y}$ and set $\boldsymbol{X}_i$ is:

$$d_{\boldsymbol{M}}(y, \boldsymbol{X}_i) = (\boldsymbol{y} - \boldsymbol{X}_i\hat{\boldsymbol{a}}_i)^T\boldsymbol{M}(\boldsymbol{y} - \boldsymbol{X}_i\hat{\boldsymbol{a}}_i). \tag{6.7}$$

The class with the minimal PSD is assigned to $\boldsymbol{y}$: $Label(\boldsymbol{y}) = \arg\min_i\{d_{\boldsymbol{M}}(\boldsymbol{y}, \boldsymbol{X}_i)\}$.

Compared with the nearest convex hull/affine hull classifier (NCH/NAH), which needs to solve c quadratic programming problems for the query sample $\boldsymbol{y}$, NSC only needs to compute a set of linear projections of $\boldsymbol{y}$ with $\boldsymbol{W}_i$, $i = 1, 2, ..., c$. Hence, NSC is much more efficient than NCH and NAH.

6.2.3 Set-to-set distance (SSD)

Given two image sets $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$, the set-to-set distance (SSD) between them can be defined as follows:

$$d(\boldsymbol{D}_1, \boldsymbol{D}_2) = \left\|\boldsymbol{D}_1\hat{\boldsymbol{a}} - \boldsymbol{D}_2\hat{\boldsymbol{b}}\right\|_2^2 \tag{6.8}$$

where $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{b}}$ can be solved by:

$$(\hat{\boldsymbol{a}}, \hat{\boldsymbol{b}}) = \arg\min_{\boldsymbol{a},\boldsymbol{b}} \|H(\boldsymbol{D}_1) - H(\boldsymbol{D}_2)\|_2^2 \tag{6.9}$$

When convex/affine/regularized constraints are imposed on the coefficient vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively, the corresponding distances are convex hull based image set distance (CHISD) [21], affine hull based image set distance (AHISD) [21] and regularized nearest points (RNP) [220], respectively. In [220], it has been shown that

l_2-norm regularized affine hull is much faster and can achieve comparable performance to convex/affine/sparse constraints. Given a linear projection matrix $\boldsymbol{P}$, the RNP model is:

$$\begin{aligned} &\min_{\boldsymbol{a},\boldsymbol{b}} \|\boldsymbol{P}(\boldsymbol{D}_1\boldsymbol{a} - \boldsymbol{D}_2\boldsymbol{b})\|_2^2 + \lambda_1 \|\boldsymbol{a}\|_2^2 + \lambda_2 \|\boldsymbol{b}\|_2^2 \\ &s.t. \sum a_i = 1, \sum b_i = 1 \end{aligned} \tag{6.10}$$

By solving Eq. (6.10), the SSD in Eq. (6.8) becomes:

$$\begin{aligned} d_{\boldsymbol{M}}(\boldsymbol{D}_1, \boldsymbol{D}_2) &= \left\|\boldsymbol{P}(\boldsymbol{D}_1\hat{\boldsymbol{a}} - \boldsymbol{D}_2\hat{\boldsymbol{b}})\right\|_2^2 \\ &= (\boldsymbol{D}_1\hat{\boldsymbol{a}} - \boldsymbol{D}_2\hat{\boldsymbol{b}})^T \boldsymbol{M}(\boldsymbol{D}_1\hat{\boldsymbol{a}} - \boldsymbol{D}_2\hat{\boldsymbol{b}}) \end{aligned} \tag{6.11}$$

In SSD based classification, given a query image set $\boldsymbol{Y}$, the SSD between it and each template set $\boldsymbol{X}_i, i = 1, 2, ..., c,$ is computed as

$$d_{\boldsymbol{M}}(\boldsymbol{Y}, \boldsymbol{X}_i) = (\boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{X}_i\hat{\boldsymbol{b}}_i)^T \boldsymbol{M}(\boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{X}_i\hat{\boldsymbol{b}}_i). \tag{6.12}$$

$\boldsymbol{Y}$ can then be classified by $Label(\boldsymbol{Y}) = l(\boldsymbol{X}_{\hat{i}})$, where $\hat{i} = \arg\min_i\{d_{\boldsymbol{M}}(\boldsymbol{Y}, \boldsymbol{X}_i)\}$.

6.3 Distance metric learning

With the definitions in Section 6.2, we can then design the metric learning algorithms for PSD and SSD based classification.

6.3.1 Point-to-set distance metric learning (PSDML)

According to Eq. (6.7), the matrix $\boldsymbol{M}$ plays a critical role in the final distance $d_{\boldsymbol{M}}(\boldsymbol{y}, \boldsymbol{X}_i)$. It is expected that a good $\boldsymbol{M}$ can be learned from the training sample sets $\{\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_c\}$, so that the PSD between a query sample $\boldsymbol{y}$ and the set $\boldsymbol{X}_{l(\boldsymbol{y})}$ can be

reduced, while the PSD between $\boldsymbol{y}$ and the other sets $\boldsymbol{X}_j$, $j \neq l(\boldsymbol{y})$, can be enlarged, where $l(\boldsymbol{y})$ is the label of $\boldsymbol{y}$.

To achieve this goal, with the given training data sets $\boldsymbol{X}_i, i = 1, 2, .., c$, we propose the following metric learning model:

$$\begin{aligned} &\min_{\boldsymbol{M}, \boldsymbol{a}_{l(\boldsymbol{x}_i)}, \boldsymbol{a}_j, \xi_{ij}^N, \xi_i^P, b} \|\boldsymbol{M}\|_F^2 + \nu(\textstyle\sum_{i,j} \xi_{ij}^N + \sum_i \xi_i^P) \\ &s.t.\ d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{X}_j) + b \geq 1 - \xi_{ij}^N, j \neq l(\boldsymbol{x}_i); \\ &\quad d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{X}_{l(\boldsymbol{x}_i)}) + b \leq -1 + \xi_i^P; \\ &\quad \boldsymbol{M} \succcurlyeq 0, \forall i, j, \xi_{ij}^N \geq 0, \xi_i^P \geq 0 \end{aligned} \tag{6.13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\boldsymbol{a}_{l(\boldsymbol{x}_i)}$ and $\boldsymbol{a}_j$ are coefficients vector for $\boldsymbol{X}_{l(\boldsymbol{x}_i)}$ and $\boldsymbol{X}_j$, b is the bias and ν is a positive constant. ξ_i^P and ξ_{ij}^N are slack variables for positive and negative pairs. $d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{X}_{l(\boldsymbol{x}_i)})$ is the PSD distance from $\boldsymbol{x}_i$ to the set it belongs to (i.e., the PSD of positive pairs), where $l(\boldsymbol{x}_i)$ is the class label of $\boldsymbol{x}_i$, and $d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{X}_j)$, $j \neq l(\boldsymbol{x}_i)$, is the PSD from $\boldsymbol{x}_i$ to other classes (i.e., the PSD of negative pairs).

Eq. (6.13) is a joint optimization problem of $\boldsymbol{M}$ and $\{\boldsymbol{a}_{l(\boldsymbol{x}_i)}, \boldsymbol{a}_j\}$. Like the strategy adopted in many multi-variable optimization problems, we minimize Eq. (6.13) by optimizing $\boldsymbol{M}$ and $\{\boldsymbol{a}_{l(\boldsymbol{x}_i)}, \boldsymbol{a}_j\}$ alternatively. When $\boldsymbol{M}$ is fixed, $\{\boldsymbol{a}_{l(\boldsymbol{x}_i)}, \boldsymbol{a}_j\}$ are solved for all the training samples. Note that here the"leave-one-out" strategy is used to compute $\boldsymbol{a}_{l(\boldsymbol{x}_i)}$. That is, $\bar{\boldsymbol{X}}_{l(\boldsymbol{x}_i)}$ is the training sample set of class $l(\boldsymbol{x}_i)$ but excluding sample $\boldsymbol{x}_i$. Then the positive pairs are formed as $(\boldsymbol{x}_i, \bar{\boldsymbol{X}}_{l(\boldsymbol{x}_i)}\hat{\boldsymbol{a}}_{l(\boldsymbol{x}_i)})$ and the negative pairs are formed as $(\boldsymbol{x}_i, \boldsymbol{X}_{j, j\neq l(\boldsymbol{x}_i)}\hat{\boldsymbol{a}}_{j, j\neq l(\boldsymbol{x}_i)})$. We label the negative pair as "+1" and the positive pair is set as "-1".

Let us denote by $\boldsymbol{z}_i = (\boldsymbol{z}_{i1}, \boldsymbol{z}_{i2})$ a generated sample pair. The covariance matrix of the two samples in $\boldsymbol{z}_i$ is $\boldsymbol{C}_i = (\boldsymbol{z}_{i1} - \boldsymbol{z}_{i2})(\boldsymbol{z}_{i1} - \boldsymbol{z}_{i2})^T$. Suppose that we generated ns

training sample pairs, and thus we have ns covariance matrices $\boldsymbol{C}_i, i = 1, 2, ..., ns$. We label $\boldsymbol{C}_i$ as "+1" or "-1" based on the label of $\boldsymbol{z}_i$, and define the following kernel function to measure the similarity between $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$:

$$k(\boldsymbol{C}_i, \boldsymbol{C}_j) = tr(\boldsymbol{C}_i \boldsymbol{C}_j) =< \boldsymbol{C}_i, \boldsymbol{C}_j > \tag{6.14}$$

where $tr(\cdot)$ is the trace operator of a matrix and $< \cdot, \cdot >$ means the inner product of matrices.

Suppose that we have a query sample pair, denoted by $\boldsymbol{z} = (\boldsymbol{z}_1, \boldsymbol{z}_2)$. The covariance matrix of $\boldsymbol{z}$ is denoted by $\boldsymbol{C}$. We introduce the following discriminative function to judge whether $\boldsymbol{z}$ is positive or negative:

$$\begin{aligned} f(\boldsymbol{C}) &= \textstyle\sum_i \beta_i l_i k(\boldsymbol{C}_i, \boldsymbol{C}) + b \\ &= \textstyle\sum_i \beta_i l_i < \boldsymbol{C}_i, \boldsymbol{C} > + b \\ &=< \textstyle\sum_i \beta_i l_i \boldsymbol{C}_i, \boldsymbol{C} > + b \end{aligned} \tag{6.15}$$

where l_i is the label of pair $\boldsymbol{z}_i$, and β_i is a weight. Let

$$\boldsymbol{M} = \textstyle\sum_i \beta_i l_i \boldsymbol{C}_i. \tag{6.16}$$

Then we have $f(\boldsymbol{C}) =< \boldsymbol{M}, \boldsymbol{C} > +b$.

The metric learning problem in Eq. (6.13) can then be converted into the following problem:

$$\begin{aligned} &\min_{\boldsymbol{M}, b, \xi} \|\boldsymbol{M}\|_F^2 + \nu \textstyle\sum_i \xi_i \\ &s.t.\ l_i(< \boldsymbol{M}, \boldsymbol{C}_i > +b) \geq 1 - \xi_i, \xi_i \geq 0 \end{aligned} \tag{6.17}$$

The Lagrange dual problem of the metric learning problem in Eq. (6.17) is:

$$\begin{aligned} &\max_{\beta} -\tfrac{1}{2} \textstyle\sum_{i,j} \beta_i \beta_j l_i l_j k(\boldsymbol{C}_i, \boldsymbol{C}_j) + \nu \textstyle\sum_i \beta_i \\ &s.t.\ 0 \leq \beta_i \leq \mu, \textstyle\sum_i \beta_i l_i = 0 \end{aligned} \tag{6.18}$$

Obviously, the minimization in Eq. (6.18) can be easily solved by the support vector machine (SVM) solvers such as LIBSVM [**?**]. Once $\boldsymbol{\beta} = [\beta_1, ..., \beta_i, ..., \beta_{ns}]$ is obtained by solving Eq. (6.18), $\boldsymbol{M}$ can be obtained by Eq. (6.16). With $\boldsymbol{M}$, the distance between two samples $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ can be computed as:

$$\begin{aligned} d_{\boldsymbol{M}}(\boldsymbol{z}_1, \boldsymbol{z}_2) &= (\boldsymbol{z}_1 - \boldsymbol{z}_2)^T \boldsymbol{M} (\boldsymbol{z}_1 - \boldsymbol{z}_2) \\ &= tr(\boldsymbol{M}\boldsymbol{C}) =< \boldsymbol{M}, \boldsymbol{C} > \end{aligned} \tag{6.19}$$

If we further require $d_{\boldsymbol{M}}(\boldsymbol{z}_1, \boldsymbol{z}_2)$ to be a Mahalanobis distance metric, $\boldsymbol{M}$ should be semi-positive definite. Similar to Xing et al.'s MMC [205] and Globerson et al.'s MCML [62], we can compute the singular value decomposition (SVD) of $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues, and then set the negative eigenvalues in $\boldsymbol{\Lambda}$ to 0, resulting in a new diagonal matrix $\boldsymbol{\Lambda}_+$. Finally, we let $\boldsymbol{M}_+ = \boldsymbol{U}\boldsymbol{\Lambda}_+\boldsymbol{V}$ be the learned matrix.

Once $\boldsymbol{M}$ is computed, $\{\boldsymbol{a}_{l(\boldsymbol{x}_i)}, \boldsymbol{a}_j\}$ are then updated, and the $\boldsymbol{M}$ is further updated, and so on. The proposed point-to-set distance metric learning (PSDML) algorithm is summarized in Table 6.2. The PSDML can be coupled with PSD based classifiers such as NSC [31], NCH [186] and NAH [186] for classification. In this chapter, we use NSC since it is much more efficient than NCH and NAH.

6.3.2 Set-to-set distance metric learning (SSDML)

With the SSD defined in Eq. (6.8), we can also learn a matrix $\boldsymbol{M}$ from the training sample sets $\{\boldsymbol{X}_1, ..., \boldsymbol{X}_i, ..., \boldsymbol{X}_n\}$ so that the SSD between sets with the same label can be reduced, while the SSD between sets with different labels can be enlarged. The proposed set-to-set distance metric learning (SSDML) model is formulated as

Table 6.2 Algorithm of point to set distance metric learning (PSDML).

Input: $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_c]$, label l, λ and ν

Output: $\boldsymbol{M}$

```
Initialize M = I
While iteration number < num
    Compute W_i, i = 1, ..., c by Eq. (6.6);
    Construct positive and negative sample pairs;
    Solve Eq. (6.18) by SVM solver;
    Update M by Eq. (6.16);
End
```

follows:

$$\begin{aligned} &\min_{\boldsymbol{M},\boldsymbol{a}_i,\boldsymbol{a}_j,\boldsymbol{a}_k,\xi^P_{ik},\xi^P_{ik},b} \|\boldsymbol{M}\|_F^2 + \nu(\textstyle\sum_{i,k}\xi^P_{ik} + \sum_{i,j}\xi^N_{ij}) \\ &s.t.\ d_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{X}_j) + b \geq 1 - \xi^N_{ij}, l(\boldsymbol{X}_i) \neq l(\boldsymbol{X}_j); \\ &\quad d_{\boldsymbol{M}}(\boldsymbol{X}_i, \boldsymbol{X}_k) + b \leq -1 + \xi^P_{ik}, l(\boldsymbol{X}_i) = l(\boldsymbol{X}_k); \\ &\quad \boldsymbol{M} \succcurlyeq 0, \forall i, j, k, \xi^N_{ij} \geq 0, \xi^P_{ik} \geq 0 \end{aligned} \tag{6.20}$$

where $\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k$ are the coefficients vector for image sets $\boldsymbol{X}_i, \boldsymbol{X}_j, \boldsymbol{X}_k$; $l(\boldsymbol{X}_i)$ means the label of set $\boldsymbol{X}_i$, and ξ^P_{ik}, ξ^N_{ij} are the slack variables for positive pairs and negative pairs.

The principles and main procedures of SSDML are similar to the PSDML in Section 6.3.1. We solve Eq. (6.20) by optimizing $\boldsymbol{M}$ and $\{\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k\}$ alternatively. When $\boldsymbol{M}$ is fixed, $\{\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{a}_k\}$ are updated to construct positive and negative sample pairs. When the sample pairs are given, the updating of matrix $\boldsymbol{M}$ can also be converted into the problem in Eq. (6.17). The algorithm of SSDML is summarized in Table 6.3. Note that the work in [204] relies on CHISD [21] and SANP [78]. As

RNP [220] is much faster than affine/convex/sparse hull based SSD computation, we choose it to learn the Mahalanobis distance metric based on l_2-norm regularized affine hull.

Table 6.3 Algorithm of set to set distance metric learning (SSDML).

Input: Training image sets $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_n]$,
label l, λ_1, λ_2 and ν

Output: $\boldsymbol{M}$

1 Initialize $\boldsymbol{M} = \boldsymbol{I}$
2 While iteration number < *num*
3 Compute SSD for each image set $\boldsymbol{X}_i$ by Eq. (6.10);
4 Construct positive and negative sample pairs;
5 Solve Eq. (6.18) by SVM solver;
6 Update $\boldsymbol{M}$ by Eq. (6.16);
7 End

6.3.3 Discussions

There are close relationships between the proposed PSDML/SSDML and SVM. The geometric interpretation of ν-SVM is to find the closest points in two (reduced) convex hulls [18]. Given two classes $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, the SVM is to solve the following problem [**?**]:

$$\begin{aligned} &\min \|\boldsymbol{X}_1\boldsymbol{a}_1 - \boldsymbol{X}_2\boldsymbol{a}_2\|_2^2 \\ &s.t. \sum a_{1i} = 1, \sum a_{2j} = 1, 0 \leq a_{1i}, a_{2j} \leq \mu \end{aligned} \tag{6.21}$$

It can be easily found that the associated discrimination function of SVM is $f(\boldsymbol{y}) = \boldsymbol{w}^T\boldsymbol{y} + \boldsymbol{b}$, where $\boldsymbol{w} = (\boldsymbol{X}_1\boldsymbol{a}_1 - \boldsymbol{X}_2\boldsymbol{a}_2)/2$, $p = (\boldsymbol{X}_1\boldsymbol{a}_1 + \boldsymbol{X}_2\boldsymbol{a}_2)/2$, $\boldsymbol{b} = -\boldsymbol{w}^T\boldsymbol{p} = (\boldsymbol{a}_2{}^T\boldsymbol{X}_2^T\boldsymbol{X}_2\boldsymbol{a}_2 - \boldsymbol{a}_1{}^T\boldsymbol{X}_1^T\boldsymbol{X}_1\boldsymbol{a}_1)/4$.

Then we have the following observation:

$$\begin{aligned} & f(\boldsymbol{y}) = \boldsymbol{w}^T\boldsymbol{y} + \boldsymbol{b} \\ & = \frac{(\boldsymbol{X}_1\boldsymbol{a}_1 - \boldsymbol{X}_2\boldsymbol{a}_2)^T}{2}\boldsymbol{y} + \frac{\boldsymbol{a}_2{}^T\boldsymbol{X}_2^T\boldsymbol{X}_2\boldsymbol{a}_2 - \boldsymbol{a}_1{}^T\boldsymbol{X}_1^T\boldsymbol{X}_1\boldsymbol{a}_1}{4} \\ & = \frac{\|\boldsymbol{y} - \boldsymbol{X}_2\boldsymbol{a}_2\|_2^2 - \|\boldsymbol{y} - \boldsymbol{X}_1\boldsymbol{a}_1\|_2^2}{4} \\ & = \frac{d(\boldsymbol{y}, \boldsymbol{X}_2) - d(\boldsymbol{y}, \boldsymbol{X}_1)}{4} \end{aligned} \tag{6.22}$$

Hence, similar to PSD based classification, the discriminative function of SVM actually uses the distance between the test sample $\boldsymbol{y}$ and each class. If $f(\boldsymbol{y}) \geq 0$, then $\boldsymbol{y}$ belongs to the first class. If $f(\boldsymbol{y}) < 0$, then $\boldsymbol{y}$ belongs to the second class. The difference, however, lies in that PSD based classifiers (e.g., NSC, NCH and NAH) solve $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ for each test sample while SVM learns $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ from the training set by classification loss minimization and margin maximization. The conventional PSD based classifiers ignore the training label information in computing $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$. With metric learning, PSDML can further utilize the class label to learn a discriminative metric for the point-to-set distance, and thus may result in better classification performance.

For set based classification, SVM can not be directly used. Actually, given two sets, SVM considers each set as one class and the distance between two classes is used as the SSD, which corresponds to CHISD [21]. Hence, it still ignores the discriminative information in calculating SSD, and is essentially different from the proposed SSD metric learning method. Actually, SSDML tries to make SSD computation discriminant, which is similar to the works in [98, 204]

Additionally, we formulate both PSDML and SSDML as a sample pair classification problem, which can be solved by standard SVM solvers. This makes metric learning very efficient.

6.4 Experimental result and analysis

We verify the performance of PSDML and SSDML on various visual classification tasks. In Section 6.4.1, we test PSDML on gender classification, digit recognition, object categorization and face recognition, while in Section 6.4.2, we test SSDML on video-to-video based face recognition.

6.4.1 PSDML experiments

Parameter setting and competing methods

There are two parameters in PSDML, i.e., λ in Eq. (6.6) and ν in Eq. (6.17). For SSDML, there are three parameters, i.e., λ_1 and λ_2 in Eq. (6.10) and ν in Eq. (6.17). For both PSDML and SSDML, ν in Eq. (6.17) is set to the default value 1 in LIBSVM. For PSDML, λ is chosen by cross-validation on the training set. For SSDML, λ_1 and λ_2 are fixed as 0.001 and 0.1, respectively.

We compare PSDML with four state-of-the-art metric learning methods (LMNN [196], ITML [41], NCA [63] and MCML [62]), three PSD based classifiers (NSC [31], NCH [186] and NAH [186]), the classical nearest neighbor classifier (NNC) and SVM. The Matlab source codes of LMNN, ITML, NCA, and MCML are obtained from the original authors, and we used the SVM toolbox from [**?**]. We

implemented NNC, NCH, NAH and NSC. The parameters of the competing methods are tuned for their best results.

Gender classification

A non-occluded subset (14 images per subject) of the AR dataset [**?**] is used, which consists of 50 male and 50 female subjects. We use the images from the first 25 males and 25 females for training, and the remaining images for testing. The images were cropped to 60×43. PCA was used to reduce the dimension of each image to 30 and 50, respectively. The experimental results listed in Table 6.4 show that PSDML gets the highest accuracy and improves the performance of PSD based classifiers (NSC, NCH and NAH).

Table 6.4 Accuracy (%) on gender classification.

dim.	NN	NSC	NCH	NAH	SVM
30	90.6	92.1	91.1	91.7	92.1
50	90.3	93.3	91.4	84.3	91.0
dim.	LMNN	ITML	NCA	MCML	PSDML
30	91.3	90.8	91.4	90.7	**93.7**
50	91.0	90.7	91.4	92.1	**95.4**

Digit recognition

Three handwritten digit datasets, Semeion [6], USPS [85] and MNIST [104], are used here.

Semeion: The Semeion dataset [6] has 1,593 handwritten digits from around

80 persons. Each sample is a 16×16 binarized image. The recognition rate on the raw features is shown in Table 6.5. On this dataset, the performance of NSC is much better than NNC. PSDML gets a recognition accuracy of 95.9%, which is the highest among all the methods used in comparison.

Table 6.5 Accuracy (%) on Semeion.

dim.	NN	NSC	NCH	NAH	SVM
256	91.4	94.2	94.1	92.5	93.4
dim.	LMNN	ITML	NCA	MCML	PSDML
256	93.9	93.5	93.9	90.0	**95.9**

USPS: The USPS dataset includes 7,291 training and 2,007 testing images [85]. Each sample is a 16×16 image. The experimental results on three dimensions (100, 150, 256) are shown in Table 6.6. We see that the results of NNC and NSC are similar. PSDML achieves the highest accuracy on different dimensions and its performance is comparable to other state-of-the-art metric learning methods.

MNIST: The MNIST [104] dataset contains a training set of 60,000 samples and a test set of 10,000 samples. There are 10 classes of images, and the size of each image is 28×28. We randomly select 200 samples per class for training and the image dimension is reduced to 100 by PCA. Ten random experiments are conducted and the average recognition rate is shown in Table 6.7. Again, PSDML performs the best among all methods.

Table 6.6 Accuracy (%) on the USPS.

dim.	NN	NSC	NCH	NAH	SVM
100	94.9	94.3	88.2	91.8	92.3
150	94.8	94.5	89.3	91.9	92.7
256	94.6	94.3	89.7	91.8	92.7
dim.	LMNN	ITML	NCA	MCML	PSDML
100	95.2	95.0	95.1	95.2	**95.4**
150	95.2	95.1	95.0	95.1	**95.3**
256	95.0	94.9	94.8	94.9	**95.2**

Table 6.7 Accuracy (%) on MNIST.

dim.	NN	NSC	NCH	NAH	SVM
100	93.3	95.2	96.0	94.0	95.7
dim.	LMNN	ITML	NCA	MCML	PSDML
100	95.0	93.4	93.5	90.1	**96.3**

Object categorization

The 17 category OXFORD flower dataset [135] is used. It contains 17 species of flowers with 80 images for each class. The χ^2 distance matrices of seven features (i.e., HSV, HOG, SIFTint, SIFTbdy, color, shape and texture vocabularies) are directly used as the input and the experiments are conducted based on the three predefined training, validation, and test splits. We test the performance of PSDML on each feature and the results are shown in Table 6.8. From the results we see that PSDML achieves the highest accuracy on all the seven features.

Face recognition

We then test the performance of PSDML on face recognition. As in [196], the Extended Yale B database [59] is used here. In addition, the FERET database [143] is also used since the images have huge pose variations, making it a good test-bed for metric learning methods.

Extended YaleB: The Extended YaleB database contains 2,414 frontal face images of 38 persons [59]. There are about 64 images for each subject. The original images were cropped to 192×168 pixels. This database has varying illuminations and expressions. A randomly generated matrix from a zero-mean normal distribution is is used to project the face image onto a 504-dimensional vector. We randomly choose 15 samples per subject for training and the rest images are used for test. PCA is used to reduce the dimension to 50, 100 and 150, respectively. On this database, the performance of NSC is much better than NNC. Compared with NSC, PSDML improves the recognition rate by about 4% and it works much better than other metric learning methods.

Table 6.8 Accuracy (%) on the 17 category OXFORD flowerers.

Features	NN	NSC	NAH	NAH	SVM
Color	52.3±2.2	55.4±2.7	55.2±2.8	56.3±2.8	56.9±2.6
Shape	53.7±3.5	66.5±2.1	66.7±2.0	63.4±1.3	60.0±2.9
Texture	31.9±3.6	52.4±2.1	52.4±1.5	45.5±1.8	47.8±3.4
HSV	52.0±2.6	59.2±2.3	59.4±2.3	57.2±3.5	57.0±2.9
HOG	36.9±1.7	51.6±2.5	51.8±2.9	47.6±2.6	47.3±1.9
SIFTint	58.7±2.1	66.5±1.3	66.5±1.4	64.5±1.0	59.7±1.0
SIFTbdy	51.7±0.9	57.6±2.3	57.7±2.2	57.6±2.8	47.5±2.8
Features	LMNN	ITML	NCA	MCML	ISDML
Color	53.1±2.5	53.5±2.6	52.8±2.8	54.1±2.7	**58.8±4.0**
Shape	50.1±1.0	55.0±1.4	54.5±2.0	55.5±1.5	**67.8±2.0**
Texture	35.5±3.0	36.2±2.5	33.8±2.6	34.5±2.0	**55.0±1.3**
HSV	54.8±2.7	53.5±3.0	54.0±2.9	52.9±3.1	**61.6±3.2**
HOG	38.3±1.1	37.5±2.5	38.2±2.5	38.7±2.8	**55.0±5.9**
SIFTint	60.0±3.4	61.2±1.9	59.8±1.5	60.4±1.3	**69.1±1.8**
SIFTbdy	53.3±4.1	54.2±2.5	53.3±2.9	53.3±2.1	**60.6±4.0**

Table 6.9 Accuracy (%) on the Extended YaleB database.

dim.	NN	NSC	NCH	NAH	SVM
50	76.3	86.1	70.9	86.1	78.1
100	80.2	88.2	75.5	87.6	82.4
150	78.3	88.9	77.1	88.9	82.3
dim.	LMNN	ITML	NCA	MCML	ISDML
50	77.4	78.3	78.9	79.0	**90.0**
100	81.1	81.0	82.4	82.9	**92.2**
150	81.8	83.1	83.5	82.1	**93.0**

FERET: The FERET face database is a large and popular database for evaluating state-of-the-art face recognition algorithms [143]. We use a subset of the database that includes 1,400 images from 200 individuals (each has 7 images). It consists the images whose names are marked with two character strings:"ba", "bj", "bk", "bd","be","bf", "bg". This subset involves variations in facial expression, illumination, and pose. The facial portion of each image was automatically cropped based on the location of eyes and mouth, and the cropped image was resized to 60×50 pixels and further pre-processed by histogram equalization.

We randomly select four images per subject as the training set and the remaining images are used as the test set. The recognition rates are shown in Table 6.10. In this dataset, the performance of NSC is worse than NNC. This is because there are great pose variations in this subset, and thus using hull to model the image set is not suitable. By metric learning, however, the classification rate can be improved greatly. The result of PSDML is much better than LMNN, ITML, NCA and MCML,

which validates the effectiveness of our algorithm.

Table 6.10 Accuracy (%) on the FERET.

dim.	NN	NSC	NCH	NAH	SVM
50	40.5	38.9	37.6	38.9	45.8
100	48.0	42.4	41.5	42.4	59.5
150	48.8	43.7	42.6	43.7	64.6
dim.	LMNN	ITML	NCA	MCML	PSDML
50	60.0	61.5	59.5	60.5	**64.0**
100	62.7	63.8	61.6	63.3	**67.8**
150	63.5	64.8	62.0	64.5	**67.8**

Time comparison

To show the efficiency of PSDML, we compare the training time of different metric learning methods. All algorithms are run in an Intel(R) Core(TM) i7- 2600K (3.4GHz) PC. The average training time on the MNIST dataset is listed in Table 6.11. We see that PSDML is much faster than other metric learning methods. In particular, it is nearly 500 times faster than MCML.

Table 6.11 Training time (s) on the MNIST.

Methods	LMNN	ITML	NCA	MCML	PSDML
run_time	75.9	141.0	3885.1	11825.1	**24.7**

6.4.2 SSDML experiments

We then test SSDML for set-to-set based classification tasks. The benchmark YouTube Celebrities dataset is used. In this experiment, we compare SSDML with those SSD based classification methods (CHISD [21], AHISD [21], SANP [78], RNP [220], MMD [193] and MDA [190]) and set-to-set similarity based methods (MSM [206] and DCC [98]). The source codes of these methods are from the original authors and we tune the parameters for their best results.

The Youtube Celebrities [98] is a large scale video dataset for face tracking and recognition, consisting of 1,910 video sequences of 47 celebrities collected from YouTube. As the videos were captured in unconstrained environments, the recognition task becomes much more challenging due to large variations in pose, illumination and expressions. The face in each frame is detected by the Viola-Jones face detector and resized to a 30×30 grayscale image.

The intensity value is used as feature. Three video sequences per subject are selected for training and six for testing. Five-fold cross validation is used. The experiments for 50, 100, 200 frames per set are conducted. The result is shown in Table 6.12. We can see that SSDML outperforms all the other methods on different frames per set.

Table 6.12 Recognition rates on YouTube (%).

Methods	50	100	200
MSM [206]	54.8±8.7	57.4±7.7	56.7±6.9
DCC [98]	57.6±8.0	62.7±6.8	65.7±7.0
MMD [193]	57.8±6.6	62.8±6.2	64.7±6.3
MDA [190]	58.5±6.2	63.3±6.1	65.4±6.6
AHISD [21]	57.5±7.9	59.7±7.2	57.0±5.5
CHISD [21]	58.0±8.2	62.8±8.1	64.8±7.1
SANP [78]	57.8±7.2	63.1±8.0	65.6±7.9
RNP [220]	59.9±7.3	63.3±8.1	64.4±7.8
SSDML	**61.9±7.3**	**65.0±8.1**	**67.0±7.1**

6.4.3 Comparison between PSDML and DSRIC

As both DSRIC and PSDML can apply to the same classification tasks, we conduct experiments on handwritten digit recognition to compare the recognition performance and efficiency of PSDML and DSRIC. Table 6.13 and Table 6.14 show the recognition accuracy and testing time, respectively. From the results, we can see that the accuracy of PSDML is a little higher than DSRIC while DSRIC is twice faster than PSDML.

6.4.4 Combination of PSDML and DSRIC

PSDML aims to improve the discrimination ability of representation based classifiers by learning a distance metric $\boldsymbol{M}$ for all classes, i.e., $(\boldsymbol{x} - \boldsymbol{D}_k\hat{\boldsymbol{a}}_k)^T \boldsymbol{M}(\boldsymbol{x} - \boldsymbol{D}_k\hat{\boldsymbol{a}}_k)$.

Table 6.13 Recognition accuracy (%) on handwritten digit recognition

Method	USPS	MNIST
PSDML	**95.4**	**96.3**
DSRIC	94.3	95.2

Table 6.14 Testing time comparison (s) on handwritten digit recognition

Method	USPS	MNIST
PSDML	0.0002	0.0066
DSRIC	**0.0001**	**0.0038**

DSRIC learns a discrimination matrix per class for classification by introducing a discrimination representation item, i.e., $(\boldsymbol{x} - \boldsymbol{B}_k\boldsymbol{x})^T(\boldsymbol{x} - \boldsymbol{B}_k\boldsymbol{x})$. Actually we can combine the advantage of PSDML and DSRIC, i.e., $(\boldsymbol{x} - \boldsymbol{B}_k\boldsymbol{x})^T\boldsymbol{M}(\boldsymbol{x} - \boldsymbol{B}_k\boldsymbol{x})$. Firstly, instead of using NSC to generate positive/negative pairs, we can use DSRIC to get pairs for distance metric learning. Then a distance metric $\boldsymbol{M}$ is learned. We conduct experiments on different classification tasks, including gender classification, face recognition and handwritten digit recognition tasks. The experiment results are shown in Table 6.15. From the experiment result, we can see that the combination method can always achieve the highest accuracy on different classification tasks. Besides, the time complexity of the combination method is the same as PSDML. Hence, the combination of PSDML and DSRIC can lead to more robust classification in terms of both accuracy and efficiency.

Table 6.15 Recognition rates on different classification tasks

Database	NSC	DSRIC	PSDML	DSRIC+PSDML
gender	93.4	94.7	95.3	**95.3**
LFW	37.8	60.8	38.2	**61.3**
USPS	94.3	94.3	95.4	**95.4**
MNIST	95.2	95.2	96.3	**96.3**

6.5 Conclusions and discussions

We extended the point-to-point distance metric learning to point-to-set distance metric learning (PSDML) and set-to-set distance metric learning (SSDML). Positive and negative sample pairs were generated from training sample sets by computing point-to-set distance (PSD) and set-to-set distance (SSD). Each sample pair was represented by its covariance matrix and a covariance kernel based discrimination function was proposed for sample pair classification. Finally, we showed that the proposed metric learning problem can be efficiently solved by SVM solvers. Experiments on various visual classification problems demonstrated that the proposed PSDML and SSDML methods can effectively improve the performance of PSD and SSD based classification. Compared with the state-of-the-art metric learning methods such as LMNN, ITML and MCML, the proposed method can achieve better classification accuracy and is significantly faster in training.

In Chapter 4,we proposed discriminative self-representation induced classifier (DSRIC) and learn a discriminative square matrix per class for classification. In this chapter, we consider representation residual as a point to set distance and learn

a discriminative distance metric to enhance representation based classification. The objective of both methods are to introduce more discrimination information to the representation process. Actually, as shown in section 6.4.4 we can combine the advantage of DSRIC and PSDML to further improve the performance of representation based classifiers.

Chapter 7

Conclusions

7.1 Conclusions

Although lots of representation based classification models have been developed and can apply to classification tasks such as face recognition, image classification, visual tracking, action recognition, etc. In small sample size problem, image set classification tasks and large-scale classification tasks, the existing representation based classifiers may fail or can not apply. This thesis aims to address representation with small sample size, representation with big sample size, image set representation, and representation with metric learning problems.

- We proposed two models for face recognition with single sample per person, i.e., multi-scale patch based collaborative representation (MSPCRC) and local generic representation (LGR). MSPCRC utilizes patch-level representation and fuses decisions of different patch sizes by margin distribution optimization. To introduce more across-subject face variations, LGR represents

the query face patch on both the gallery patch dictionary and generic variation patch dictionary. Our extensive experiments validated that the proposed methods outperform many state-of-the-art patch based face recognition algorithms. Compared to patch based collaborative representation, LGR achieves higher accuracy but less efficiency.

- A novel feature-level self-representation concept was proposed. We developed self-representation induced classifier (SRIC) and proved that SRIC is equivalent to l_2-norm regularized nearest subspace classifier and principle component analysis with shrinkage. To improve the discrimination ability of self-representation, a discriminative SRIC (DSRIC) is developed and its time complexity is only related with feature dimension and number of classes. Hence, DSRIC can apply to classification tasks with a large amount of samples.

- A novel collaborative representation set to sets distance (CRSSD) and collaborative representation based image set classification (ISCRC) framework was proposed. Regularized affine hull and kernelized convex hull based ISCRC models were developed. ISCRC outperforms the state-of-the-art image set based face recognition method in terms of both accuracy and efficiency.

- We extended point-to-point distance metric learning to point-to-set (PSDML) and set-to-set (SSDML) distance metric learning. Both PSDML and SSDML are solved by standard support vector machine solvers and therefore can apply to large scale classification tasks. To the best of our knowledge, this is the first work for point-to-set and set-to-set distance metric learning.

7.2 Future work

This thesis has shown that there are many possibilities be explored in the extension of the developed representation based classifiers. In the future work, we will focus on the following directions:

- Different from the existing sample-level representation, in this thesis we proposed feature-level self-representation and developed the corresponding classifiers. We will combine sample-level and feature-level representation together and develop two-dimensional representation based classification models.

- Dictionary learning can get a compact and discriminative representation by learning a set of bases. For face recognition with single sample per person, we will learn a local generic variation dictionary. For image set based face recognition, we will consider to learn an image set dictionary.

- There are multi-modal and cross-modal tasks in computer vision and pattern recognition. We will extend the point-to-set and set-to-set metric learning algorithms to cross-modal and multi-modal tasks. Additionally, there are a large number of unlabeled samples in real-world applications, which can help learn a distance metric with better generalization ability. Hence, semi-supervised metric learning will be taken into account as well.

- Deep learning has attracted much attention and achieved great success in computer vision tasks. We will combine deep learning and the representation based classification models together to improve the recognition performance.

Bibliography

[1] Vitaly Ablavsky and Stan Sclaroff. Learning parameterized histogram kernels on the simplex manifold for image and action classification. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1473–1480. IEEE, 2011.

[2] Forest Agostinelli, Michael R Anderson, and Honglak Lee. Robust image denoising with multi-column deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1493–1501, 2013.

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. -svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.

[4] Nasir Ahmed, T Natarajan, and Kamisetty R Rao. Discrete cosine transform. *Computers, IEEE Transactions on*, 100(1):90–93, 1974.

[5] Ognjen Arandjelovic, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell. Face recognition with image sets using manifold density divergence. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 581–588. IEEE, 2005.

[6] Arthur Asuncion and David J Newman. Uci machine learning repository,

2007.

[7] Francis R Bach, Gert RG Lanckriet, and Michael I Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the twenty-first international conference on Machine learning*, page 6. ACM, 2004.

[8] Ronen Basri, Tal Hassner, and Lihi Zelnik-Manor. Approximate nearest subspace search with applications to pattern recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[9] Ronen Basri and David W Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.

[10] Gaston Baudat and Fatiha Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

[11] Peter N. Belhumeur, João P Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.

[12] Kristin P Bennett and Erin J Bredensteiner. Duality and geometry in svm classifiers. In *Machine Learning, International Conference on*, pages 57–64. ACM, 2000.

[13] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, 2003.

[14] John Blitzer, Kilian Q Weinberger, and Lawrence K Saul. Distance metric

learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[15] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[16] Matthew Brand. Morphable 3d models from video. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–456. IEEE, 2001.

[17] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[18] D Burges and C Crisp. A geometric interpretation of v−svm classifiers. In *Advances in Neural Information Processing Systems*, pages 244–250, 2000.

[19] John Y Campbell. *The econometrics of financial markets*. princeton University press, 1997.

[20] Rui Caseiro, Pedro Martins, João F Henriques, Fátima Silva Leite, and Jorge Batista. Rolling riemannian manifolds to solve the multi-class classification problem. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 41–48. IEEE, 2013.

[21] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2567–2573. IEEE, 2010.

[22] Hakan Cevikalp, Bill Triggs, and Robi Polikar. Nearest hyperdisk methods

for high-dimensional classification. In *Proceedings of the 25th international conference on Machine learning*, pages 120–127. ACM, 2008.

[23] Hakan Cevikalp, Bill Triggs, Hasan Serhan Yavuz, Yalçın Küçük, Mahide Küçük, and Atalay Barkana. Large margin classifiers based on affine hulls. *Neurocomputing*, 73(16):3160–3168, 2010.

[24] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[25] Hwann-Tzong Chen, Huang-Wei Chang, and Tyng-Luh Liu. Local discriminant embedding and its variants. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 846–853. IEEE, 2005.

[26] Ning Chen, Jun Zhu, and Eric P Xing. Predictive subspace learning for multi-view data: a large margin approach. In *Advances in neural information processing systems*, pages 361–369, 2010.

[27] S. Chen, J. Liu, and Z.H. Zhou. Making flda applicable to face recognition with one sample per person. *Pattern recognition*, 37(7):1553–1555, 2004.

[28] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[29] Yi-Chen Chen, Vishal M Patel, P Jonathon Phillips, and Rama Chellappa. Dictionary-based face recognition from video. In *Computer Vision–ECCV 2012*, pages 766–779. Springer, 2012.

[30] Yuejie Chi and Fatih Porikli. Connecting the dots in multi-class classifica-

tion: From nearest subspace to collaborative representation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3602–3609. IEEE, 2012.

[31] Jen-Tzung Chien and Chia-Chen Wu. Discriminant waveletfaces and nearest feature classifiers for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(12):1644–1649, 2002.

[32] Wen-Sheng Chu, Ju-Chin Chen, and Jenn-Jier James Lien. Kernel discriminant transformation for image set-based face recognition. *Pattern Recognition*, 44(8):1567–1580, 2011.

[33] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Unsupervised metric learning for face identification in tv video. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1559–1566. IEEE, 2011.

[34] Adam Coates and Andrew Y Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 921–928, 2011.

[35] Thomas F Coleman and Yuying Li. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058, 1996.

[36] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[37] Camille Couprie, Clément Farabet, Laurent Najman, and Yann LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arX-*

iv:1301.3572, 2013.

[38] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[39] K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin analysis of the lvq algorithm. *Advances in neural information processing systems*, pages 479–486, 2003.

[40] Zhen Cui, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Image sets alignment for video-based face recognition. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2626–2633. IEEE, 2012.

[41] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216. ACM, 2007.

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[43] Weihong Deng, Jiani Hu, and Jun Guo. Extended src: Undersampled face recognition via intraclass variant dictionary. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1864–1870, 2012.

[44] David L Donoho and Iain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995.

[45] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least

angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[46] Kjersti Engan, Sven Ole Aase, and JH Husoy. Frame based signal compression using method of optimal directions (mod). In *Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on*, volume 4, pages 1–4. IEEE, 1999.

[47] Wei Fan and Dit-Yan Yeung. Locally linear models on face appearance manifolds with application to dual-subspace based classification. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1384–1390. IEEE, 2006.

[48] LMJ Florack, BM Ter Haar Romeny, Jan J Koenderink, and Max A Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4(2):171–187, 1994.

[49] William T. Freeman and Edward H Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern analysis and machine intelligence*, 13(9):891–906, 1991.

[50] Jerome H. Friedman, Jon Luis Bentley, and Raphael Ari Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematics Software*, 3(3):209–226, September 1977.

[51] Kazuhiro Fukui, Bjorn Stenger, and Osamu Yamaguchi. A framework for 3d object recognition using the kernel constrained mutual subspace method. In *Computer Vision–ACCV 2006*, pages 315–324. Springer, 2006.

[52] Keinosuke Fukunaga and Patrenahalli M Narendra. A branch and bound algorithm for computing k-nearest neighbors. *Computers, IEEE Transactions on*, 100(7):750–753, 1975.

[53] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Springer, 2012.

[54] Q. Gao, L. Zhang, and D. Zhang. Face recognition using flda with single training image per person. *Applied Mathematics and Computation*, 205(2):726–734, 2008.

[55] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Kernel sparse representation for image classification and face recognition. In *Computer Vision–ECCV 2010*, pages 1–14. Springer, 2010.

[56] Shenghua Gao, Ivor Wai-Hung Tsang, and Liang-Tien Chia. Sparse representation with kernels. *Image Processing, IEEE Transactions on*, 22(2):423–434, 2013.

[57] Shenghua Gao, IW.-H. Tsang, and Yi Ma. Learning category-specific dictionary and shared dictionary for fine-grained image categorization. *Image Processing, IEEE Transactions on*, 23(2):623–634, Feb 2014.

[58] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel. Extracting deep bottleneck features using stacked auto-encoders. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3377–3381. IEEE, 2013.

[59] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):643–660, 2001.

[60] R. Gilad Bachrach, A. Navot, and N. Tishby. Margin based feature selection-theory and algorithms. In *Proceedings of the twenty-first international con-*

ference on Machine learning, page 43. ACM, 2004.

[61] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, volume 99, pages 518–529, 1999.

[62] Amir Globerson and Sam Roweis. Metric learning by collapsing classes. In *NIPS 2006*.

[63] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *NIPS 2004*.

[64] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[65] R. Gross and J. Shi. The cmu motion of body (mobo) database. *Technical Report*, 27(1):1–13, 2001.

[66] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 498–505. Ieee, 2009.

[67] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV 2010*.

[68] Asela Gunawardana and William Byrne. Convergence theorems for generalized alternating minimization procedures. *The Journal of Machine Learning Research*, 6:2049–2073, 2005.

[69] Abdenour Hadid and M Pietikainen. From still image to video-based face recognition: an experimental analysis. In *Automatic Face and Gesture Recognition, IEEE Conference on*, pages 813–818. IEEE, 2004.

[70] Chris Harris and Mike Stephens. A combined corner and edge detector. In

Alvey vision conference, volume 15, page 50. Manchester, UK, 1988.

[71] Li He, Hairong Qi, and Russell Zaretzki. Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 345–352. IEEE, 2013.

[72] Ran He, Tieniu Tan, Liang Wang, and Wei-Shi Zheng. $l_{2,1}$-regularized correntropy for robust feature selection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2504–2511. IEEE, 2012.

[73] Ran He, Wei-Shi Zheng, and Bao-Gang Hu. Maximum correntropy criterion for robust face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1561–1576, 2011.

[74] Ran He, Wei-Shi Zheng, Tieniu Tan, and Zhenan Sun. Half-quadratic-based iterative minimization for robust sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):261–275, 2014.

[75] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):328–340, 2005.

[76] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[77] David W Hosmer, Stanley Lemeshow, and Rodney X Sturdivant. *Introduction to the logistic regression model*. Wiley Online Library, 2000.

[78] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 121–128. IEEE, 2011.

[79] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Face recognition using sparse approximated nearest points between image sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1992–2004, 2012.

[80] De-An Huang and Yu-Chiang Frank Wang. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2496–2503. IEEE, 2013.

[81] De-An Huang and Yu-Chiang Frank Wang. With one look: robust face recognition using single sample per person. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 601–604. ACM, 2013.

[82] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[83] Zhiwu Huang, Xiaowei Zhao, Shiguang Shan, Ruiping Wang, and Xilin Chen. Coupling alignments with recognition for still-to-video face recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3296–3303. IEEE, 2013.

[84] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[85] Jonathan J. Hull. A database for handwritten text recognition research. *TPAMI*, 16(5):550–554, 1994.

[86] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2005.

[87] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 73–80. IEEE, 2013.

[88] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, 2013.

[89] Nan Jiang, Wenyu Liu, and Ying Wu. Order determination and sparsity-regularized metric learning adaptive visual tracking. In *CVPR 2012*.

[90] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1697–1704. IEEE, 2011.

[91] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, 2013.

[92] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *Computer Vision-ECCV 2004*, pages 228–241. Springer, 2004.

[93] Meina Kan, Shiguang Shan, Yu Su, Dong Xu, and Xilin Chen. Adaptive discriminant learning for face recognition. *Pattern Recognition*, 46(9):2497–

2509, 2013.

[94] Dor Kedem, Stephen Tyree, Kilian Weinberger, Fei Sha, and Gert Lanckriet. Non-linear metric learning. In *NIPS 2012*.

[95] Minyoung Kim, Sanjiv Kumar, Vladimir Pavlovic, and Henry Rowley. Face tracking and recognition with visual constraints in real-world videos. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8. IEEE, 2008.

[96] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606–617, 2007.

[97] Tae-Kyun Kim and Josef Kittler. Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):318–327, 2005.

[98] T.K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):1005–1018, 2007.

[99] Shu Kong and Donghui Wang. A dictionary learning approach for classification: separating the particularity and the commonality. In *Computer Vision–ECCV 2012*, pages 186–199. Springer, 2012.

[100] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[101] Ritwik Kumar, Arunava Banerjee, and Baba C Vemuri. Volterrafaces: Dis-

criminant analysis using volterra kernels. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 150–155. IEEE, 2009.

[102] Ritwik Kumar, Arunava Banerjee, Baba C. Vemuri, and Hanspeter Pfister. Maximizing all margins: Pushing face recognition with kernel plurality. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2375 –2382, nov. 2011.

[103] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[104] Y. LeCun and C. Cortes. The mnist database of handwrittn digits. `http://yann.lecun.com/exdb/mnist/`.

[105] Kuang-Chih Lee, Jeffrey Ho, Ming-Hsuan Yang, and David Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 313–320. IEEE, 2003.

[106] Yuchun Lee. Handwritten digit recognition using k nearest-neighbor, radial-basis function, and backpropagation neural networks. *Neural computation*, 3(3):440–449, 1991.

[107] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[108] Jing Li and Nigel M Allinson. A comprehensive review of current local

features for computer vision. *Neurocomputing*, 71(10):1771–1787, 2008.

[109] D. Lin and X. Tang. Recognize high resolution faces: From macrocosm to microcosm. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1355–1362. IEEE, 2006.

[110] Weifeng Liu, Puskal P Pokharel, and José C Príncipe. Correntropy: properties and applications in non-gaussian signal processing. *Signal Processing, IEEE Transactions on*, 55(11):5286–5298, 2007.

[111] Yiguang Liu, Shuzhi Sam Ge, Chunguang Li, and Zhisheng You. k-ns: A classifier by the distance to the nearest subspace. *Neural Networks, IEEE Transactions on*, 22(8):1256–1268, 2011.

[112] Canyi Lu, Jinhui Tang, Min Lin, Liang Lin, Shuicheng Yan, and Zhouchen Lin. Correntropy induced l2 graph for robust subspace clustering. In *Proc. 14th IEEE International Conf. Computer Vision (ICCV)*, December 2013. in press.

[113] Jiwen Lu, Yap-Peng Tan, and Gang Wang. Discriminative multimanifold analysis for face recognition from a single training sample per person. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):39–51, 2013.

[114] Juwei Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, 26(2):181–191, 2005.

[115] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*,

34(4):791–804, 2012.

[116] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[117] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.

[118] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

[119] Stéphane Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[120] Benoit B Mandelbrot. How long is the coast of britain. *Science*, 156(3775):636–638, 1967.

[121] S Marčelja. Mathematical description of the responses of simple cortical cells*. *JOSA*, 70(11):1297–1300, 1980.

[122] Aleix M Martínez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(6):748–763, 2002.

[123] A.M. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[124] Xue Mei and Haibin Ling. Robust visual tracking using &# x2113; 1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443. IEEE, 2009.

[125] Alexis Mignon and Frédéric Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *CVPR 2012*.

[126] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[127] Pabitra Mitra, CA Murthy, and Sankar K. Pal. Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):301–312, 2002.

[128] Hoda Mohammadzade and Dimitrios Hatzinakos. Projection into expression subspaces for face recognition from single sample per person. *Affective Computing, IEEE Transactions on*, 4(1):69–82, 2013.

[129] Gianluca Monaci, Philippe Jost, Pierre Vandergheynst, Boris Mailhe, Sylvain Lesage, and Rémi Gribonval. Learning multimodal dictionaries. *Image Processing, IEEE Transactions on*, 16(9):2272–2283, 2007.

[130] Hans P Moravec. Rover visual obstacle avoidance. In *IJCAI*, pages 785–790, 1981.

[131] 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-11)*, pages 689–696, 2011.

[132] Hien V Nguyen, Vishal M Patel, Nasser M Nasrabadi, and Rama Chellappa. Sparse embedding: A framework for sparsity promoting dimensionality reduction. In *Computer Vision–ECCV 2012*, pages 414–427. Springer, 2012.

[133] Urs Niesen, Devavrat Shah, and Gregory W Wornell. Adaptive alternating minimization algorithms. *Information Theory, IEEE Transactions on*, 55(3):1423–1429, 2009.

[134] Mila Nikolova and Michael K Ng. Analysis of half-quadratic minimization

methods for signal and image recovery. *SIAM Journal on Scientific computing*, 27(3):937–966, 2005.

[135] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *CVPR 2006*.

[136] Masashi Nishiyama, Mayumi Yuasa, Tomoyuki Shibata, Tomokazu Wakasugi, Tomokazu Kawahara, and Osamu Yamaguchi. Recognizing faces of moving people by hierarchical image-set matching. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8. IEEE, 2007.

[137] Haydemar Núñez, Cecilio Angulo, and Andreu Català. Rule extraction from support vector machines. In *ESANN*, pages 107–112, 2002.

[138] Bruno A Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[139] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[140] Shibin Parameswaran and Kilian Q Weinberger. Large margin multi-task metric learning. In *Advances in neural information processing systems*, pages 1867–1875, 2010.

[141] Vishal M Patel, Tao Wu, Soma Biswas, P Jonathon Phillips, and Rama Chellappa. Dictionary-based face recognition under variable lighting and pose. *Information Forensics and Security, IEEE Transactions on*, 7(3):954–965, 2012.

[142] Xinjun Peng and Yifei Wang. Geometric algorithms to large margin clas-

sifier based on affine hulls. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(2):236–246, 2012.

[143] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *TPAMI*, 22(10):1090–1104, 2000.

[144] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, volume 1, pages 947–954. IEEE, 2005.

[145] Norman Poh, Chi Ho Chan, Josef Kittler, Sébastien Marcel, Christopher McCool, Enrique Argones Rúa, José Luis Alba Castro, Mauricio Villegas, Roberto Paredes, Vitomir Struc, et al. An evaluation of video-to-video face verification. *Information Forensics and Security, IEEE Transactions on*, 5(4):781–801, 2010.

[146] J.R. Quinlan. *C4. 5: programs for machine learning*. Morgan Kaufmann, 1993.

[147] Ignacio Ramirez, Pablo Sprechmann, and Guillermo Sapiro. Classification and clustering via dictionary learning with structured incoherence and shared features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3501–3508. IEEE, 2010.

[148] J.B. Ramsey. Tests for specification errors in classical linear least-squares regression analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 31(2):350–371, 1969.

[149] M Ranzato and Geoffrey E Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2551–2558. IEEE, 2010.

[150] Sarunas J Raudys and Anil K. Jain. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3):252–264, 1991.

[151] L. Reyzin and R.E. Schapire. How boosting the margin can also boost classifier complexity. In *Proceedings of the 23rd international conference on Machine learning*, pages 753–760. ACM, 2006.

[152] Brian Ripley. Classification and regression trees. *R package version*, pages 1–0, 2005.

[153] R.T. Rockafellar. *Convex analysis*, volume 28. Princeton Univ Pr, 1997.

[154] Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In *Subspace, Latent Structure and Feature Selection*, pages 34–51. Springer, 2006.

[155] S. Rosset, J. Zhu, and T. Hastie. Boosting as a regularized path to a maximum margin classifier. *The Journal of Machine Learning Research*, 5:941–973, 2004.

[156] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[157] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008.

[158] Ruslan Salakhutdinov and Geoffrey Hinton. An efficient learning procedure for deep boltzmann machines. *Neural computation*, 24(8):1967–2006, 2012.

[159] Frederik Schaffalitzky and Andrew Zisserman. Multi-view matching for unordered image sets, or how do i organize my holiday snaps?. In *Computer VisionłECCV 2002*, pages 414–431. Springer, 2002.

[160] R.E. Schapire, Y. Freund, P. Bartlett, and W.S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

[161] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 936. John Wiley & Sons, 2012.

[162] Hae Jong Seo and Peyman Milanfar. Face verification using the lark representation. *Information Forensics and Security, IEEE Transactions on*, 6(4):1275–1286, 2011.

[163] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3626–3633. IEEE, 2013.

[164] Thomas Serre, Lior Wolf, and Tomaso Poggio. Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 994–1000. IEEE, 2005.

[165] Shiguang Shan, Bo Cao, Wen Gao, and Debin Zhao. Extended fisherface for face recognition from a single example image per person. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 2,

pages II–81. IEEE, 2002.

[166] J. Shawe-Taylor and N. Cristianini. Robust bounds on generalization from the margin distribution. In *4th European Conference on Computational Learning Theory*. Citeseer, 1999.

[167] John Shawe-Taylor and Nello Cristianini. Margin distribution bounds on generalization. In *Computational Learning Theory*, pages 263–273. Springer, 1999.

[168] C. Shen and H. Li. Boosting through optimization of margin distributions. *Neural Networks, IEEE Transactions on*, 21(4):659–666, 2010.

[169] C. Shen and H. Li. On the dual formulation of boosting algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2216–2231, 2010.

[170] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 129–136, 2011.

[171] Johannes Stallkamp, Hazim Kemal Ekenel, and Rainer Stiefelhagen. Video-based face recognition on real-world data. In *Computer Vision, IEEE International Conference on*, pages 1–8. IEEE, 2007.

[172] Y. Su, S. Shan, X. Chen, and W. Gao. Hierarchical ensemble of global and local classifiers for face recognition. *Image Processing, IEEE Transactions on*, 18(8):1885–1896, 2009.

[173] Y. Su, S. Shan, X. Chen, and W. Gao. Adaptive generic learning for face recognition from a single sample per person. In *Computer Vision and Pattern*

Recognition (CVPR), 2010 IEEE Conference on, pages 2699–2706. IEEE, 2010.

[174] X. Tan, S. Chen, Z.H. Zhou, and F. Zhang. Recognizing partially occluded, expression variant faces from single training image per person with som and soft k-nn ensemble. *Neural Networks, IEEE Transactions on*, 16(4):875–886, 2005.

[175] Xiaoyang Tan, Songcan Chen, Zhi-Hua Zhou, and Fuyan Zhang. Face recognition from a single image per person: A survey. *Pattern Recognition*, 39(9):1725–1745, 2006.

[176] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[177] Sebastian Thrun. Extracting rules from artificial neural networks with distributed representations. *Advances in neural information processing systems*, pages 505–512, 1995.

[178] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[179] Lorenzo Torresani, Danny B Yang, Eugene J Alexander, and Christoph Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–493. IEEE, 2001.

[180] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces.

In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[181] Tinne Tuytelaars and Luc Van Gool. Matching widely separated views based on affine invariant regions. *International journal of computer vision*, 59(1):61–85, 2004.

[182] Shimon Ullman and Ronen Basri. Recognition by linear combinations of models. *IEEE transactions on pattern analysis and machine intelligence*, 13(10):992–1006, 1991.

[183] T. Van Gestel, J.A.K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine Learning*, 54(1):5–32, 2004.

[184] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 2. Wiley New York, 1998.

[185] Thomas Vetter. Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28(2):103–116, 1998.

[186] P. Vincent and Y. Bengio. K-local hyperplane and convex distance nearest neighbor algorithms. In *Advances in Neural Information Processing Systems*, 2002.

[187] P. Viola and M.J. Jones. Robust real-time face detection. *Interational Journal of Computer Vision*, 57(2):137–154, 2004.

[188] Faqiang Wang, Wangmeng Zuo, Lei Zhang, Deyu Meng, and David Zhang. A kernel classification framework for metric learning. *arXiv:1309.5823*, 2013.

[189] Jie Wang, Kostas N Plataniotis, Juwei Lu, and Anastasios N Venetsanopou-

los. On solving the face recognition problem with one training sample per subject. *Pattern recognition*, 39(9):1746–1762, 2006.

[190] Ruiping Wang and Xilin Chen. Manifold discriminant analysis. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 429–436. IEEE, 2009.

[191] Ruiping Wang, Huimin Guo, Larry S Davis, and Qionghai Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2496–2503. IEEE, 2012.

[192] Ruiping Wang, Shiguang Shan, Xilin Chen, Qionghai Dai, and Wen Gao. Manifold-manifold distance and its application to face recognition with image sets. *Image Processing, IEEE Transactions on*, 21(10):4466–4479, 2012.

[193] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–8. IEEE, 2008.

[194] Shenlong Wang, D Zhang, Yan Liang, and Quan Pan. Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2216–2223. IEEE, 2012.

[195] Zhaowen Wang, Jianchao Yang, Nasser Nasrabadi, and Thomas Huang. A max-margin perspective on sparse representation-based classification. In *Computer Vision, IEEE International Conference on*. IEEE, 2013.

[196] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric

learning for large margin nearest neighbor classification. In *NIPS 2006*.

[197] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. *Computer Vision–ACCV 2009*, pages 88–97, 2010.

[198] L. Wolf, T. Hassner, and Y. Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(10):1978–1990, 2011.

[199] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 529–534. IEEE, 2011.

[200] John Wright, Arvind Ganesh, Allen Yang, Zihan Zhou, and Yi Ma. Sparsity and robustness in face recognition. *arXiv preprint arXiv:1111.1014*, 2011.

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

[202] Lei Wu, Rong Jin, and Anil K Jain. Tag completion for image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(3):716–727, 2013.

[203] Yang Wu, Michihiko Minoh, and Masayuki Mukunoki. Collaboratively regularized nearest points for set based recognition. In *Proceedings of the British Machine Vision Conference*, pages 1–10. BMVA Press, 2013.

[204] Yang Wu, Michihiko Minoh, Masayuki Mukunoki, and Shihong Lao. Set based discriminative ranking for recognition. In *Computer Vision–ECCV 2012*, pages 497–510. Springer, 2012.

[205] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *NIPS 2002*.

[206] Osamu Yamaguchi, Kazuhiro Fukui, and K-i Maeda. Face recognition using temporal image sequence. In *Automatic Face and Gesture Recognition, IEEE Conference on*, pages 318–323. IEEE, 1998.

[207] S. Yan, D. Xu, B. Zhang, H.J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40–51, 2007.

[208] A.Y. Yang, A. Ganesh, Z. Zhou, S.S. Sastry, and Y. Ma. A review of fast l1-minimization algorithms for robust face recognition. *Arxiv preprint arXiv:1007.3753*, 2010.

[209] A.Y. Yang, S.S. Sastry, A. Ganesh, and Y. Ma. Fast l1-minimization algorithms and an application in robust face recognition: A review. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 1849–1852. IEEE, 2010.

[210] Jian Yang, Delin Chu, Lei Zhang, Yong Xu, and Jingyu Yang. Sparse representation classifier steered discriminative projection with applications to face recognition. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(7):1023–1035, 2013.

[211] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–

8. IEEE, 2008.

[212] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[213] Meng Yang, Luc Van Gool, , and Lei Zhang. Sparse variation dictionary learning for face recognition with a single training sample per person. In *Proc. 14th IEEE International Conf. Computer Vision (ICCV)*, December 2013. in press.

[214] Meng Yang, D Zhang, and Jian Yang. Robust sparse coding for face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 625–632. IEEE, 2011.

[215] Meng Yang, Lei Zhang, Xiangchu Feng, and David Zhang. Fisher discrimination dictionary learning for sparse representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 543 –550, nov. 2011.

[216] Meng Yang, Lei Zhang, S.C.-K. Shiu, and D. Zhang. Robust kernel representation with statistical local features for face recognition. *Neural Networks and Learning Systems, IEEE Transactions on*, 24(6):900–912, 2013.

[217] Meng Yang, Lei Zhang, Jian Yang, and D. Zhang. Regularized robust coding for face recognition. *Image Processing, IEEE Transactions on*, 22(5):1753–1766, 2013.

[218] Meng Yang, Lei Zhang, Jian Yang, and David Zhang. Metaface learning for sparse representation based face recognition. In *Image Processing, IEEE International Conference on*, pages 1601–1604. IEEE, 2010.

[219] Meng Yang, Lei Zhang, David Zhang, and Shenlong Wang. Relaxed collaborative representation for pattern classification. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 2224–2231. IEEE, 2012.

[220] Meng Yang, Pengfei Zhu, Luc Van Gool, and Lei Zhang. Face recognition based on regularized nearest points between image sets. In *Automatic Face and Gesture Recognition, IEEE Conference on*, pages 1–7. IEEE, 2013.

[221] Kai Yu, Yuanqing Lin, and John Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1713–1720. IEEE, 2011.

[222] Zhou Yu, Fei Wu, Yi Yang, Qi Tian, Jiebo Luo, and Yueting Zhuang. Discriminative coupled dictionary hashing for fast cross-media retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 395–404. ACM, 2014.

[223] Xiao-Tong Yuan, Xiaobai Liu, and Shuicheng Yan. Visual classification with multitask joint sparse representation. *Image Processing, IEEE Transactions on*, 21(10):4349–4360, 2012.

[224] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Robert Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[225] J. Zhang, Y. Yan, and M. Lades. Face recognition: eigenface, elastic matching, and neural nets. *Proceedings of the IEEE*, 85(9):1423–1435, 1997.

[226] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: Which helps face recognition? In *Int. Conf. on Comput. Vis,*

2011.

[227] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010.

[228] Wei Zhang, Xiaogang Wang, and Xiaoou Tang. Coupled information-theoretic encoding for face photo-sketch recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 513–520. IEEE, 2011.

[229] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *Acm Computing Surveys (CSUR)*, 35(4):399–458, 2003.

[230] Ji Zhu, Saharon Rosset, Trevor Hastie, and Robert Tibshirani. 1-norm support vector machines. In *Advances in neural information processing systems*, pages 49–56, 2003.

[231] Pengfei Zhu, Lei Zhang, Qinghua Hu, and Simon CK Shiu. Multi-scale patch based collaborative representation for face recognition with margin distribution optimization. In *Computer Vision–ECCV 2012*, pages 822–835. Springer, 2012.

[232] Pengfei Zhu, Wangmeng Zuo, Lei Zhang, S.C.-K. Shiu, and D. Zhang. Image set-based collaborative representation for face recognition. *Information Forensics and Security, IEEE Transactions on*, 9(7):1120–1132, July 2014.