Jun-Bao Li · Shu-Chuan Chu
Jeng-Shyang Pan

Kernel Learning Algorithms for Face Recognition

Springer

Kernel Learning Algorithms for Face Recognition

Jun-Bao Li · Shu-Chuan Chu
Jeng-Shyang Pan

Kernel Learning Algorithms for Face Recognition

Springer

Jun-Bao Li
Harbin Institute of Technology
Harbin
People's Republic of China

Shu-Chuan Chu
Flinders University of South Australia
Bedford Park, SA
Australia

Jeng-Shyang Pan
HIT Shenzhen Graduate School
Harbin Institute of Technology
Shenzhen City
Guangdong Province
People's Republic of China

ISBN 978-1-4614-0160-5 ISBN 978-1-4614-0161-2 (eBook)
DOI 10.1007/978-1-4614-0161-2
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013944551

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Face recognition (FR) is an important research topic in the pattern recognition area and is widely applied in many areas. Learning-based FR achieves good performance, but linear learning methods share their limitations on extracting the features of face image, change of pose, illumination, and express causing the image to present a complicated nonlinear character. The recently proposed kernel method is regarded an effective method for extracting the nonlinear features and is widely used. Kernel learning is an important research topic in the machine learning area, and some theory and applications fruits are achieved and widely applied in pattern recognition, data mining, computer vision, image, and signal processing. The nonlinear problems are solved at large with kernel function and system performances such as recognition accuracy and prediction accuracy that are largely increased. However, the kernel learning method still endures a key problem, i.e., kernel function and its parameter selection. Research has shown that kernel function and its parameters have a direct influence on data distribution in the nonlinear feature space, and inappropriate selection will influence the performance of kernel learning. Research on self-adaptive learning of kernel function and its parameter has important theoretical value for solving the kernel selection problem widely endured by the kernel learning machine, and has the same important practical meaning for improvement of kernel learning systems.

The main contributions of this book are described as follows:

First, for parameter selection problems endured by kernel learning algorithms, this dissertation proposes the kernel optimization method with the data-dependent kernel. The definition of data-dependent kernel is extended, and its optimal parameters are achieved through solving the optimization equation created based on Fisher criterion and maximum margin criterion. Two kernel optimization algorithms are evaluated and analyzed from two different views.

Second, for problems of computation efficiency and storage space endured by kernel learning-based image feature extraction, an image matrix-based Gaussian kernel directly dealing with the images is proposed. The image matrix need not be transformed to the vector when the kernel is used in image feature extraction. Moreover, by combining the data-dependent kernel and kernel optimization, we propose an adaptive image matrix-based Gaussian kernel which not only directly deals with the image matrix but also adaptively adjusts the parameters of the

kernels according to the input image matrix. This kernel can improve the performance of kernel learning-based image feature extraction.

Third, for the selection of kernel function and its parameters endured by traditional kernel discriminant analysis, the data-dependent kernel is applied to kernel discriminant analysis. Two algorithms named FC+FC-based adaptive kernel discriminant analysis and MMC+FC-based adaptive kernel discriminant analysis are proposed. The algorithms are based on the idea of combining kernel optimization and linear projection-based two-stages algorithm. The algorithms adaptively adjust the structure of kernels according to the distribution of the input samples in the input space and optimize the mapping of sample data from the input space to the feature space. Thus the extracted features have more class discriminative ability compared with traditional kernel discriminant analysis. As regards parameter selection problem endured by traditional kernel discriminant analysis, this report presents the Nonparametric Kernel Discriminant Analysis (NKDA) method which solves the performance of classifier owing to unfitted parameter selection. As regards kernel function and its parameter selection, kernel structure self-adaptive discriminant analysis algorithms are proposed and tested with simulations.

Fourth, for problems endured by the recently proposed Locality Preserving Projection (LPP) algorithm: (1) The class label information of training samples is not used during training; (2) LPP is a linear transformation-based feature extraction method and is not able to extract the nonlinear features; (3) LPP endures the parameter selection problem when it creates the nearest neighbor graph. For the above problems, this dissertation proposes a supervised kernel locality preserving projection algorithm, and the algorithm applies the supervised no parameters method for creating the nearest neighbor graph. The extracted nonlinear features have the largest class discriminative ability. The improved algorithm solves the above problems endured by LPP and enhances its performance on feature extraction.

Fifth, for Pose, Illumination and Expression (PIE) problems endured by image feature extraction for face recognition, three kernel learning-based face recognition algorithms are proposed. (1) To make full use of advantages of signal processing and learning-based methods on image feature extraction, a face image extraction method of combining Gabor wavelet and enhanced kernel discriminant analysis is proposed. (2) Polynomial kernel is extended to fractional power polynomial model, and is used for kernel discriminant analysis. A fraction power polynomial model-based kernel discriminant analysis for feature extraction of facial image is proposed. (3) In order to make full use of the linear and nonlinear features of images, an adaptively fusing PCA and KPCA for face image extraction is proposed.

Finally, on the training samples number and kernel function and their parameter endured by Kernel Principal Component Analysis, this report presents a one-class support vector-based Sparse Kernel Principal Component Analysis (SKPCA). Moreover, data-dependent kernel is introduced and extended to propose SKPCA algorithm. First, a few meaningful samples are found for solving the constraint optimization equation, and these training samples are used to compute the kernel

matrix which decreases the computing time and saving space. Second, kernel optimization is applied to self-adaptive, adjusting the data distribution of the input samples and the algorithm performance is improved based on the limit training samples.

The main contents of this book include Kernel Optimization, Kernel Sparse Learning, Kernel Manifold Learning, Supervised Kernel Self-adaptive Learning, and Applications of Kernel Learning.

Kernel Optimization

This book aims to solve parameter selection problems endured by kernel learning algorithms, and presents kernel optimization method with the data-dependent kernel. The book extends the definition of data-dependent kernel and applies it to kernel optimization. The optimal structure of the input data is achieved through adjusting the parameter of data-dependent kernel for high class discriminative ability for classification tasks. The optimal parameter is achieved through solving the optimization equation created based on Fisher criterion and maximum margin criterion. Two kernel optimization algorithms are evaluated and analyzed from two different views. On practical applications, such as image recognition, for problems of computation efficiency and storage space endured by kernel learning-based image feature extraction, an image matrix-based Gaussian kernel directly dealing with the images is proposed in this book. Matrix Gaussian kernel-based kernel learning is implemented on image feature extraction using image matrix directly without transforming the matrix into vector for the traditional kernel function. Combining the data-dependent kernel and kernel optimization, this book presents an adaptive image matrix-based Gaussian kernel with self-adaptively adjusting the parameters of the kernels according to the input image matrix, and the performance of image-based system is largely improved with this kernel.

Kernel Sparse Learning

On the training samples number and kernel function and its parameter endured by Kernel Principal Component Analysis; this book presents one-class support vector-based Sparse Kernel Principal Component Analysis (SKPCA). Moreover, data-dependent kernel is introduced and extended to propose SKPCA algorithm. First, the few meaningful samples are found with solving the constraint optimization equation, and these training samples are used to compute the kernel matrix which decreases the computing time and saving space. Second, kernel optimization is applied to self-adaptive adjusting data distribution of the input samples and the algorithm performance is improved based on the limit training samples.

Kernel Manifold Learning

On the nonlinear feature extraction problem endured by Locality Preserving Projection (LPP) based manifold learning, and this book proposes a supervised kernel locality preserving projection algorithm creating the nearest neighbor graph. The extracted nonlinear features have the largest class discriminative ability, and it solves the above problems endured by LPP and enhances its performance on feature extraction. This book presents kernel self-adaptive manifold learning. The traditional unsupervised LPP algorithm is extended to the supervised and kernelized learning. Kernel self-adaptive optimization solves kernel function and its parameters selection problems of supervised manifold learning, which improves the algorithm performance on feature extraction and classification.

Supervised Kernel Self-Adaptive Learning

On parameter selection problem endured by traditional kernel discriminant analysis, this book presents Nonparametric Kernel Discriminant Analysis (NKDA) to solve the performance of classifier owing to unfitted parameter selection. On kernel function and its parameter selection, kernel structure self-adaptive discriminant analysis algorithms are proposed and tested with simulations. For the selection of kernel function and its parameters endured by traditional kernel discriminant analysis, the data-dependent kernel is applied to kernel discriminant analysis. Two algorithms named FC+FC-based adaptive kernel discriminant analysis and MMC+FC-based adaptive kernel discriminant analysis are proposed. The algorithms are based on the idea of combining kernel optimization and linear projection-based two-stage algorithm. The algorithms adaptively adjust the structure of kernels according to the distribution of the input samples in the input space and optimize the mapping of sample data from the input space to the feature space. Thus the extracted features have more class discriminative ability compared with traditional kernel discriminant analysis.

Acknowledgements

This work is supported by the National Science Foundation of China under Grant no. 61001165, the HIT Young Scholar Foundation of the 985 Project, and the Fundamental Research Funds for the Central Universities Grant No. HIT.BRETIII.201206

Contents

Chapter 1
Introduction

1.1 Basic Concept

Face recognition (FR) has become a popular research topic in the computer vision, image processing, and pattern recognition areas. Recognition performance of the practical FR system is largely influenced by the variations in illumination conditions, viewing directions or poses, facial expression, aging, and disguises. FR provides the wide applications in commercial, law enforcement, and military, and so on, such as airport security and access control, building surveillance and monitoring, human–computer intelligent interaction and perceptual interfaces, smart environments at home, office, and cars. An excellent FR method should consider what features are used to represent a face image and how to classify a new face image based on this representation. Current feature extraction methods can be classified into signal processing and statistical learning methods. On signal-processing-based methods, feature-extraction-based Gabor wavelets are widely used to represent the face image, because the kernels of Gabor wavelets are similar to two-dimensional receptive field profiles of the mammalian cortical simple cells, which capture the properties of spatial localization, orientation selectivity, and spatial frequency selectivity to cope with the variations in illumination and facial expressions. On the statistical-learning-based methods, the dimension reduction methods are widely used. In this book, we have more attentions on learning-based FR method. In the past research, the current methods include *supervised learning, unsupervised learning, and semi-supervised learning*.

1.1.1 Supervised Learning

Supervised learning is a popular learning method through mapping the input data into the feature space, and it includes classification and regression. During learning the mapping function, the sample with the class labels is used to training. Many works discuss supervised learning extensively including pattern recognition, machine learning.

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_1,

Supervised learning methods are consisted of two kinds of generative or discriminative methods. Generative models assume that the data that are independently and identically distributed are subject to one probability density function, for example, posteriori estimation (MAP) [1], empirical Bayes, and variational Bayes [2]. Different from data generation process, discriminative methods directly make the decision boundary of the classes. The decision boundary is represented with a parametric function of data through minimizing the classification error on the training set [1]. Empirical risk minimization (ERM) is a widely adopted principle in discriminative supervised learning, for example, neural networks [3] and logistic regression [2]. As opposed to probabilistic methods, the decision boundary is modeled directly, which overcomes structural risk minimization (SRM) principle by Vapnik's [4], and this method adds a regularity criterion to the empirical risk. So that, the classifier has a good generalization ability. Most of the above classifiers implicitly or explicitly require the data to be represented as a vector in a suitable vector space [5].

Ensemble classifiers are used to combine multiple component classifiers to obtain a meta-classifier, for example, bagging [6] and boosting [7, 8]. Bagging is a short form for bootstrap aggregation, which trains multiple instances of a classifier on different subsamples. Boosting samples trains data more intelligently, and it is difficult for the existing ensemble to classify with a higher preference.

1.1.2 Unsupervised Learning

Unsupervised learning is a significantly more difficult problem than classification. Many clustering algorithms have already been proposed [9], and we broadly divide the clustering algorithms into groups. As an example of a sum of squared error (SSE) minimization algorithm, K-means is the most popular and widely used clustering algorithm. K-means is initialized with a set of random cluster centers, for example, ISODATA [10], linear vector quantization [11].

Parametric mixture models are widely used in machine learning areas [12], for example, GMM [13, 14] has been extensively used for clustering. Since it assumes that each component is homogeneous, unimodal, and generated using a Gaussian density, its performance is limited. For that, an improved method called latent Dirichlet allocation [15] was proposed, as a multinomial mixture model. Several mixture models have been extended to their nonparametric form by taking the number of components to infinity [16–18]. Spectral clustering algorithms [19–21] are popular nonparametric models, and it minimizes an objective function. Kernel K-means is a related kernel-based algorithm, which generalizes the Euclidean-distance-based K-means to arbitrary metrics in the feature space. Using the kernel trick, the data are first mapped into a higher-dimensional space using a possibly nonlinear map, and a K-means clustering is performed in the higher-dimensional space.

1.1.3 Semi-Supervised Algorithms

Semi-supervised learning methods attempt to improve the performance of a supervised or an unsupervised learning in the presence of side information. This side information can be in the form of unlabeled samples in the supervised case or pair-wise constraints in the unsupervised case. An earlier work by Robbins and Monro [22] on sequential learning can also be viewed as related to semi-supervised learning, for example, Vapnik's overall risk minimization (ORM) principle [23]. Usually, the underlying geometry of the data is captured by representing the data as a graph, with samples as the vertices, and the pair-wise similarities between the samples as edge weights. Several graph-based algorithms such as label propagation [24], Markov random walks [25], graph cut algorithms [26], spectral graph transducer [27], and low-density separation [28]. The second assumption is cluster assumption [29]. Many successful semi-supervised algorithms are TSVM [30] and semi-supervised SVM [31]. These algorithms assume a model for the decision boundary, resulting in a classifier.

1.2 Kernel Learning

Kernel method was firstly proposed in Computational Learning Theory Conference in 1992. In the conference, support vector machine (SVM) theory was introduced and caused the large innovation of machine learning. The key technology of SVM is that the inner product of the nonlinear vector is defined with kernel function. Based on the kernel function, the data are mapped into high-dimensional feature space with kernel mapping. Many kernel learning methods were proposed through kernelizing the linear learning methods.

Kernel learning theory is widely paid attentions by researchers, and kernel learning method is successfully applied in pattern recognition, regression estimation, and so on [32–38].

1.2.1 Kernel Definition

Kernel function defines the nonlinear inner product $<(x), (y)>$ of the vector x and y, then

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle \tag{1.1}$$

The definition is proposed based on Gram matrix and positive matrix.

1. Kernel construction

Kernel function is a crucial factor for influencing the kernel learning, and different kernel functions cause the different generalization of kernel learning, such as SVM. Researchers construct different kernel in the different application. In current kernel learning algorithms, polynomial kernel, Gaussian kernel, and sigmoid kernel and RBF kernel are popular kernel, as follows.

Polynomial kernel

$$k(x,y) = (x \cdot y)^d \quad (d \in N) \tag{1.2}$$

Gaussian kernel

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (\sigma > 0) \tag{1.3}$$

Sigmoid kernel

$$k(x,z) = \tanh(\alpha\langle x,z\rangle + \beta),\ (\alpha > 0, \beta < 0) \tag{1.4}$$

RBF kernel

$$k(x,z) = \exp(-\rho d(x,z)),\ (\rho > 0) \tag{1.5}$$

where $d(x,z)$ is a distance measure.

1.2.2 Kernel Character

Based on the definition of kernel, it seems that it firstly constructs the nonlinear mapping space and then computes the inner product of the input vectors in the nonlinear mapping space. In fact, in the practical application, kernel function represents the nonlinear mapping space. Based on this idea, the inner product computation can avoid the computation in the nonlinear mapping space. So it need not know the mapping equation in advanced in the practical application. In fact, for one kernel function, it can construct the kernel-based feature space, where the inner product is defined by kernel function. The vector in the nonlinear mapping space can described as follows.

(a) Vector norm:

$$\|\Phi(x)\|_2 = \sqrt{\|\Phi(x)\|^2} = \sqrt{\langle \Phi(x), \Phi(x)\rangle} = \sqrt{k(x,x)} \tag{1.6}$$

(b) Vector linear combination norm:

$$\begin{aligned}\left\|\sum_{i=1}^{l}\alpha_i\Phi(x_i)\right\|^2 &= \left\langle\sum_{i=1}^{l}\alpha_i\Phi(x_i),\sum_{j=1}^{l}\alpha_j\Phi(x_j)\right\rangle \\ &= \sum_{i=1}^{l}\alpha_i\sum_{j=1}^{l}\alpha_j\langle\Phi(x_i),\Phi(x_j)\rangle\sum_{i=1}^{l}\alpha_i\sum_{j=1}^{l}\alpha_j k(x_i,x_j)\end{aligned} \tag{1.7}$$

(c) Norm of two vectors differ

$$\begin{aligned}\|\Phi(x)-\Phi(z)\|_2 &= \langle\Phi(x)-\Phi(z),\Phi(x)-\Phi(z)\rangle \\ &= \langle\Phi(x),\Phi(x)\rangle - 2\langle\Phi(x),\Phi(z)\rangle + \langle\Phi(x),\Phi(z)\rangle \\ &= k(x,x)-2k(x,z)+k(z,z)\end{aligned} \tag{1.8}$$

According to T. M. Cover's pattern classification theory, one complicate pattern classification will be more easily classified in the higher-dimensional mapping space than low-dimensional nonlinear mapping space.

Suppose that k is real positive definite kernel, and $R^{\mathbb{R}} := \{f : \mathbb{R} \to R\}$ is the kernel mapping space from $\mathbb{R}$ to R, then the mapping from $\mathbb{R}$ to $R^{\mathbb{R}}$ is defined by

$$\begin{aligned}\Phi : \mathbb{R} &\to R^{\mathbb{R}} \\ x &\to k(\cdot,x)\end{aligned} \tag{1.9}$$

is Reproduced kernel mapping.

Mercer proposition given the function k in $\mathbb{R}^2$, then

$$T_k : H_2(\mathbb{R}) \to H_2(\mathbb{R})$$

$$(T_k f)(x) := \int_{\mathbb{R}} k(x,x')f(x')\mathrm{d}\mu(x') \tag{1.10}$$

is positive, that is, for all $f \in H_2(\mathbb{R}^2)$, then

$$\int_{\mathbb{R}^2} k(x,x')f(x)f(x')\mathrm{d}\mu(x)\mu(x') \geq 0 \tag{1.11}$$

Then

$$k(x,x') = \sum_{j=1}^{n_f}\lambda_j\psi_j(x)\psi_j(x') \tag{1.12}$$

Suppose that k satisfies kernel with Mercer Proposition, define the mapping from $\mathbb{R}$ to $R^{\mathbb{R}}$ as

$$\Phi : \mathbb{R} \to h_2^{n_f}$$

$$x \to \left(\sqrt{\lambda_j}\psi_j(x)\right)_{j=1,2,\ldots,n_f} \tag{1.13}$$

is Mercer kernel mapping, where $\psi_j \in H_2(\mathbb{R})$ denotes the eigenvalue function T_k and its eigenvalue λ_j, n_f and ψ_j have the same definition to Mercer Proposition.

Supposed that k is Mercer kernel, Φ is Mercer kernel mapping, for all $(x, x') \in \mathbb{R}^2$, then

$$\langle \Phi(x), \Phi(x') \rangle = k(x, x') \tag{1.14}$$

Mercer kernel mapping is used to construct the Hilbert space, and the inner product is defined with kernel function. Mercer kernel and position definite kernel may be defined with the inner product in Hilbert kernel.

Suppose that $\mathbb{R} = [a, c]$ is compact region, $k : [a, c] \times [a, c] \to C$ is continuous function, then k is position definite kernel, only each continuous function $f : \mathbb{R} \to C$, then

$$\int_{\mathbb{R}^2} k(x, x') f(x) f(x') \mathrm{d}x \mathrm{d}x' \geq 0. \tag{1.15}$$

1.3 Current Research Status

In 1960s, kernel function has been introduced into pattern recognition, but it just developed to one hot research topic until SVM was successfully used in pattern recognition areas [39, 40]. In the following research, Scholkopf introduced kernel learning into feature extraction [41–43] and proposed kernel principal component analysis (KPCA) [41, 42], Mika [44–47], Baudat [48] and Roth [49] extended the linear discriminant analysis (LDA) method into kernel version through using kernel trick. From that, kernel learning and its relative research attracted researchers' interest, and three research stages of the kernel learning are shown as follows. In the first stage, before 2000, the beginning research of the kernel learning, the main research fruits include KPCA, kernel discriminant analysis. Other few research fruits were achieved. In the second stage, 2000–2004, some relative kernel learning algorithms are achieved such as kernel HMM [50], kernel associative memory [51]. This stage of research is regarded as the basis for the following research on kernel learning.

In the third stage, from 2005 to now, many researchers devote their interests to the kernel learning research area. They developed many kernel learning methods and applied them to many practical applications. Many universities and research institutions carried out kernel research study earlier, such as Yale, MIT, Microsoft Corporation, and achieved fruitful results. China's Shanghai Jiao Tong University,

Nanjing University, Nanjing University, Harbin Institute of Technology, Shenzhen Graduate Institute, and other research institutions have recently carried out learning algorithms and applications of kernel research gradually and have achieved some results.

Although research on kernel learning only lasted about a decade, but it has formed a relatively complete system of kernel learning research and a number of research branches are developed. They are kernel-based classification, kernel clustering algorithms, feature extraction based on kernel learning algorithms, kernel-based learning neural networks and kernel applications and other research application branch.

1.3.1 Kernel Classification

Kernel learning method originated in SVMs [39, 40, 52], which is a typical classification algorithms. In subsequent studies, the researchers made a variety of kernel-based learning classification algorithm. Peng et al. applied kernel method to improve the nearest neighbor classifier [53], and implemented the nearest neighbor classification in the nonlinear mapping space. Recently, researchers have proposed some new kernel-based learning classification algorithm, Jiao et al. [54] proposed kernel matching pursuit classifier (KMPC) algorithm, as well as Zhang et al. [55] proposed a learning-based minimum distance classifier, and to optimize the kernel function parameters applied to the idea of algorithm design, the algorithm can automatically adjust the parameters of the kernel function and enhance the ability of nonlinear classification problem. In addition, Jiao et al. [56] proposed kernel matching pursuit classification algorithm.

1.3.2 Kernel Clustering

Kernel clustering algorithm was developed only in recent years as an important branch of kernel learning. Ma et al. [57] proposed a discriminant analysis based on kernel clustering algorithm, which is the main idea to use kernel learning to map the original data into a high-dimensional feature space. This method performed C-means clustering discriminant analysis algorithms. Have et al. [58] use kernel methods of spectral clustering method extended to kernel spectral clustering method, and Kima [59] and other researcher presented other various kernel-based learning methods for clustering comparison. Kernel clustering-applied research also received the majority of attention of scholars. Researchers use kernel clustering as target tracking, character recognition, and other fields. Studies have shown that kernel learning algorithm clustering has been successfully applied and is widely used in various fields.

1.3.3 Kernel Feature Extraction

The branch of study and research in the kernel learning field was the most active research topic. Kernel feature extraction algorithm to learn a wealth of linear feature extraction algorithm can learn from the research results, coupled with its wide range of applications prompted the research branch of rapid development. Most of the algorithm is a linear feature extraction algorithm expansion, and improvement of the algorithm is a landmark KPCA algorithm [42] and KFD algorithm [49]. The success of these two algorithms with the kernel method to solve linear principal component analysis (PCA) and LDA in dealing with highly complex nonlinear distribution structure classification problem encountered difficulties. In subsequent research work, Yang et al. proposed KPCA algorithm for FR. Facial feature extraction based on combined Fisherface algorithm is also presented [60]. The combined Fisherface method extract two different characteristics of an object using PCA and KPCA respectively, and the two different characteristics are complementary to the image recognition. Finally the combination of two characteristics is used to identify the image. Liu [61] extended the polynomial kernel function as fractional power polynomial models and combined with KPCA and Gabor wavelet for FR. Lu and Liang et al. proposed kernel direct discriminant analysis (KDDA) algorithm [62–64], and the method differed from traditional KDA algorithms. Liang et al. [65] and Liang [66] proposed two kinds of criterions to solve mapping matrix. Their algorithms are similar because they all solved eigenvectors of degree minimum mapping matrix between unrelated and related original samples. This method was reported good results on recognition. Yang [67] analyzed theoretically KFD algorithm connotation and proposed a two stages of KPCA + LDA for KFD algorithm, and Belhumeur et al. [68] proposed the improved Fisherface algorithm in order to solve the SSS problem. Yang et al. in subsequent work theoretically proved the rationality of the algorithm [69]. In addition, Baudat et al. [70] proposed that a kernel-based generalized discriminant analysis algorithm with KDA difference is that it found that change in interclass matrix is zero matrix such a transformation matrix. Zheng and other researchers proposed a weighting factor based on the maximum interval discriminant analysis algorithm [71], and Wu et al. proposed a fuzzy kernel discriminant analysis algorithm [72], and Tao et al. proposed KDDA algorithmic improvements [73]. In fact, the choice of kernel parameter has a great impact on the performance of the algorithm. In order to try to avoid the kernel function parameters on the algorithm, the researchers applied kernel parameter selection method into KDA algorithm to improve the performance. For example, Huang [74], Wang [75], and Chen [76] selected kernel function parameters to improve KDA, and other improved KDA algorithms are presented by other literatures [77–86]. In addition, many other kernel-based learning methods were presented for feature extraction and classification. Wang et al. proposed a kernel-based HMM algorithm [87]. Yang used the kernel method to independent component analysis (ICA) for feature extraction and presented kernel independence Element Analysis (KICA) [88], and Chang et al. [89] proposed

kernel Particle filter for target tracking. Zhang et al. [90] proposed Kernel Pooled Local Subspaces for feature extraction and classification.

1.3.4 Kernel Neural Network

In recent years, kernel method was applied to neural networks. For example, Shi et al. [91] will reproduce kernel and organically combine it with neural networks to propose reproduced kernel neural networks. The classical application of kernel in the neural network is self-organizing map (SOM) [60, 92–94]. The goal of SOM is to use low-dimensional space of the original high-dimensional space point that represents the point of making this representation to preserve the original distance or similarity relation. Zhang et al. [91] proposed kernel associative memory combined with wavelet feature extraction algorithm. Zhang et al. [95] proposed a Gabor wavelet associative memory combined with the kernel-based FR algorithm, Sussner et al. [96] proposed based on dual kernel associative memory algorithms, and Wang et al. [97] used the empirical kernel map associative memory to enhance the performance of the algorithm method.

1.3.5 Kernel Application

With the kernel research, kernel learning methods are widely used in many applications, for example, character recognition [98, 99], FR [100–102], text classification [103, 104], DNA analysis [105–107], expert system [108], image retrieval [109]. Kernel-learning-based FR is the most popular application, and kernel method provides one solution to PIE problems of FR.

1.4 Problems and Contributions

Kernel learning is an important research topic in the machine learning area, and some theory and application fruits are achieved and widely applied in pattern recognition, data mining, computer vision, and image and signal processing areas. The nonlinear problems are solved at large with kernel function and system performances such as recognition accuracy, prediction accuracy largely increased. However, kernel learning method still endures a key problem, i.e., kernel function and its parameter selection. Researches show that kernel function and its parameters have the direct influence on the data distribution in the nonlinear feature space, and the inappropriate selection will influence the performance of kernel learning. Research on self-adaptive learning of kernel function and its parameter has an important theoretical value for solving the kernel selection problem widely

endured by kernel learning machine and has the same important practical meaning for the improvement of kernel learning systems.

The main contributions of this book are described as follows.

Firstly, for the parameter selection problems endured by kernel learning algorithms, the book proposes kernel optimization method with the data-dependent kernel. The definition of data-dependent kernel is extended, and the optimal parameters of data-dependent kernel are achieved through solving the optimization equation created based on Fisher criterion and maximum margin criterion. Two kernel optimization algorithms are evaluated and analyzed from two different views.

Secondly, for the problems of computation efficiency and storage space endured by kernel-learning-based image feature extraction, an image-matrix-based Gaussian kernel directly dealing with the images is proposed. The image matrix is not needed to be transformed to the vector when the kernel is used in image feature extraction. Moreover, by combining the data-dependent kernel and kernel optimization, we propose an adaptive image-matrix-based Gaussian kernel which not only directly deals with the image matrix but also adaptively adjusts the parameters of the kernels according to the input image matrix. This kernel can improve the performance of kernel-learning-based image feature extraction.

Thirdly, for the selection of kernel function and its parameters endured by traditional kernel discriminant analysis, the data-dependent kernel is applied to kernel discriminant analysis. Two algorithms named FC + FC-based adaptive kernel discriminant analysis and MMC + FC-based adaptive kernel discriminant analysis are proposed. Two algorithms are based on the idea of combining kernel optimization and linear projection based on two-stage algorithm. Two algorithms adaptively adjust the structure of kernels according to the distribution of the input samples in the input space and optimize the mapping of sample data from the input space to the feature space. So the extracted features have more class discriminative ability compared with traditional kernel discriminant analysis. On parameter selection problem endured by traditional kernel discriminant analysis, this report presents nonparameter kernel discriminant analysis (NKDA) and this method solves the performance of classifier owing to unfitted parameter selection. On kernel function and its parameter selection, kernel structure self-adaptive discriminant analysis algorithms are proposed and testified with the simulations.

Fourthly, for the problems endured by the recently proposed locality preserving projection (LPP) algorithm, (1) the class label information of training samples is not used during training; (2) LPP is a linear-transformation-based feature extraction method and is not able to extract the nonlinear features; (3) LPP endures the parameter selection problem when it creates the nearest neighbor graph. For the above problems, this book proposes a supervised kernel LPP algorithm, and this algorithm applies the supervised no parameter method of creating the nearest neighbor graph. The extracted nonlinear features have the largest class discriminative ability. The improved algorithm solves the above problems endured by LPP and enhances its performance on feature extraction.

Fifthly, for the pose, illumination, and expression (PIE) problems endured by image feature extraction for FR, three kernel-learning-based FR algorithms are proposed. (1) In order to make full use of advantages of signal-processing- and learning-based methods on image feature extraction, a face image extraction method of combining Gabor wavelet and enhanced kernel discriminant analysis is proposed. (2) Polynomial kernel is extended to fractional power polynomial model, and it is used to kernel discriminant analysis. A fraction power-polynomial-model-based kernel discriminant analysis for feature extraction of facial image is proposed. (3) In order to make full use of the linear and nonlinear features of images, an adaptively fusing PCA and KPCA for face image extraction are proposed.

Sixthly, on the training samples' number and kernel function and its parameter endured by KPCA, this report presents one-class support-vector-based sparse kernel principal component analysis (SKPCA). Moreover, data-dependent kernel is introduced and extended to propose SKPCA algorithm. Firstly, the few meaningful samples are found with solving the constraint optimization equation, and these training samples are used to compute the kernel matrix which decreases the computing time and saving space. Secondly, kernel optimization is applied self-adaptively to adjust the data distribution of the input samples and the algorithm performance is improved based on the limit training samples.

1.5 Contents of this Book

The main contents of this book include kernel optimization, kernel sparse learning, kernel manifold learning, supervised kernel self-adaptive learning, and applications of kernel learning.

Kernel Optimization

This research aims to solve the parameter selection problems endured by kernel learning algorithms and presents kernel optimization method with the data-dependent kernel. This research extends the definition of data-dependent kernel and applies it to kernel optimization. The optimal structure of the input data is achieved through adjusting the parameter of data-dependent kernel for high class discriminative ability for classification tasks. The optimal parameter is achieved through solving the optimization equation created based on Fisher criterion and maximum margin criterion. Two kernel optimization algorithms are evaluated and analyzed from two different views. On the practical applications, such as image recognition, for the problems of computation efficiency and storage space endured by kernel-learning-based image feature extraction, an image-matrix-based Gaussian kernel directly dealing with the images is proposed in this research. Matrix Gaussian-kernel-based kernel learning is implemented on image feature extraction using image matrix directly without transforming the matrix into vector for the traditional kernel function. Combining the data-dependent kernel and kernel optimization, this research presents an adaptive image–matrix-based

Gaussian kernel with self-adaptively adjusting the parameters of the kernels according to the input image matrix, and the performance of image-based system is largely improved with this kernel.

Kernel Sparse Learning

On the training samples' number and kernel function and its parameter endured by KPCA, this research presents one-class support-vector-based SKPCA. Moreover, data-dependent kernel is introduced and extended to propose SKPCA algorithm. Firstly, the few meaningful samples are found with solving the constraint optimization equation, and these training samples are used to compute the kernel matrix which decreases the computing time and saving space. Secondly, kernel optimization is applied self-adaptively to adjust the data distribution of the input samples and the algorithm performance is improved based on the limit training samples.

Kernel Manifold Learning

On the nonlinear feature extraction problem endured by LPP-based manifold learning, this research proposes a supervised kernel LPP algorithm with supervised, creating the nearest neighbor graph. The extracted nonlinear features have the largest class discriminative ability, and it solves the above problems endured by LPP and enhances its performance on feature extraction. This research presents kernel self-adaptive manifold learning. The traditional unsupervised LPP algorithm is extended to the supervised and kernelized learning. Kernel self-adaptive optimization solves kernel function and its parameter selection problems of supervised manifold learning, which improves the algorithm performance on feature extraction and classification.

Supervised Kernel Self-Adaptive Learning

On parameter selection problem endured by traditional kernel discriminant analysis, this research presents NKDA to solve the performance of classifier owing to unfitted parameter selection. On kernel function and its parameter selection, kernel structure self-adaptive discriminant analysis algorithms are proposed and testified with the simulations. For the selection of kernel function and its parameters endured by traditional kernel discriminant analysis, the data-dependent kernel is applied to kernel discriminant analysis. Two algorithms named FC + FC-based adaptive kernel discriminant analysis and MMC + FC-based adaptive kernel discriminant analysis are proposed. Two algorithms are based on the idea of combining kernel optimization and linear projection based on two-stage algorithm. Two algorithms adaptively adjust the structure of kernels according to the distribution of the input samples in the input space and optimize the mapping of sample data from the input space to the feature space. So the extracted features have more class discriminative ability compared with traditional kernel discriminant analysis.

References

1. Duda RO, Hart PE, Stork DG (2000) Pattern classification. Wiley-Interscience Publication, New York
2. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
3. Bishop C (2005) Neural networks for pattern recognition. Oxford University Press, Oxford
4. Vapnik V (1982) Estimation of dependences based on empirical data. Springer, New York
5. Tan P, Steinbach M, Kumar V (2005) Introduction to data mining. Pearson Addison Wesley, Boston
6. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
7. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of the international conference on machine learning, pp 148–156
8. Freund Y (1995) Boosting a weak learning algorithm by majority. Inf Comput 121(2):256–285
9. Berkhin P (2006) A survey of clustering data mining techniques, Chap. 2. Springer, pp 25–71
10. Ball G, Hall D (1965) ISODATA, a novel method of data analysis and pattern classification. Technical Report NTIS AD 699616, Stanford Research Institute, Stanford, CA
11. Lloyd S (1982) Least squares quantization in pcm. IEEE Trans Inf Theory 28:129–137
12. McLachlan GL, Basford KE (1987) Mixture models: inference and applications to clustering. Marcel Dekker
13. McLachlan GL, Peel D (2000) Finite mixture models. Wiley
14. Figueiredo M, Jain A (2002) Unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell, pp 381–396
15. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
16. Teh Y, Jordan M, Beal M, Blei D (2006) Hierarchical Dirichlet processes. J Am Stat Assoc 101(476):1566–1581
17. Blei DM, Jordan MI (2004) Hierarchical topic models and the nested Chinese restaurant process. Adv Neural Inf Process Syst
18. Rasmussen CE (2000) The infinite gaussian mixture model. Adv Neural Inf Process Syst, pp 554–560
19. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell 22:888–905
20. Ng AY, Jordan MI, Weiss Y (2001) On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst, pp 849–856
21. Li Z, Liu J, Chen S, Tang X (2007) Noise robust spectral clustering. In: Proceedings of the international conference on computer vision, pp 1–8
22. Robbins H, Monro S (1951) A stochastic approximation method. Ann Math Stat 22:400–407
23. Vapnik V, Sterin A (1977) On structural risk minimization or overall risk in a problem of pattern recognition. Autom Remote Control 10(3):1495–1503
24. Bengio Y, Alleau OB, Le Roux N (2006) Label propagation and quadratic criterion. In: Chapelle O, Schölkopf B, Zien A (eds) Semi-supervised learning. MIT Press, pp 193–216
25. Szummer M, Jaakkola T (2001) Partially labeled classification with Markov random walks. Adv Neural Inf Process Syst, pp 945–952
26. Blum A, Chawla S (2001) Learning from labeled and unlabeled data using graph mincuts. In: Proceedings of the international conference on machine learning, pp 19–26
27. Joachims T (2003) Transductive learning via spectral graph partitioning. In: Proceedings of the international conference on machine learning, pp 290–297
28. Chapelle O, Zien A (2005) Semi-supervised classification by low density separation. In: Proceedings of the international conference on artificial intelligence and statistics, pp 57–64
29. Chapelle O, Scholkopf B, Zien A (eds) (2006) Semi-supervised learning. MIT Press

30. Joachims T (1999) Transductive inference for text classification using support vector machines. In: Proceedings of the international conference on machine learning, pp 200–209
31. Fung G, Mangasarian O (2001) Semi-supervised support vector machines for unlabeled data classification. Optim Methods Softw 15:29–44
32. Muller KR, Mika S, Ratsch G, Tsuda K, Scholkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 12(2):181–201
33. Campbell C (2002) Kernel methods: a survey of current techniques. Neurocomputing 48:63–84
34. Ruiz A, Lopez-de-Teruel PE (2001) Nonlinear Kernel-based statistical pattern analysis. IEEE Trans Neural Networks 12(1):1045–1052
35. Mika S, Ratsch G, Weston J, Scholkopf B, Muller K (1999) Fisher discriminant analysis with kernels. IEEE neural networks for signal processing workshop, pp 41–48
36. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12:2385–2404
37. Scholkopf B, Smola A, Muller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
38. Mika S, Ratsch G, Weston J, Scholkopf B, Smola A, Muller KR (2003) Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature space. IEEE Trans Pattern Anal Mach Intell 25(5):623–628
39. Vapnik VN (2000) The nature of statistical learning theory, 2nd edn. Springer, NY
40. Vapnik VN (1995) The nature of statistical learning theory. Springer
41. Scholkopf B, Smola A, Muller KR (1996) Nonlinear component analysis as a kernel eigenvalue problem. Technical Report No. 44. Max-Planck-Institut fur biologische Kybernetik, Tubingen Neural Computation 10(5):1299–1319
42. Scholkopf B, Smola A, Muller KR (1997) Kernel principal component analysis. In: Gerstner W (ed) Artificial neural networks, pp 583–588
43. Scholkopf B, Mika S, Burges CJC, Knirsch P, Muller KR, Ratsch G, Smola AJ (1999) Input space vs. feature space in kernel-based methods. IEEE Trans Neural Networks 10(5):1000–1017
44. Mika S, Ratsch G, Weston J, Scholkopf B, Smola A, Muller KR (2003) Constructing descriptive and discriminative non-linear feature: Rayleigh coefficients in kernel feature space. IEEE Trans Pattern Anal Mach Intell 25(5):623–628
45. Mika S, Ratsch G, Muller KR (2001) A mathematical programming approach to the Kernel Fisher algorithm. In Leen TK, Dietterich TG, Tresp V (eds) Advances in neural information processing systems. MIT Press
46. Mika S, Smola A, Scholkopf B (2001) An improved training algorithm for kernel fisher discriminants. In: Jaakkola T, Richardson T (eds) Proceedings AISTATS, pp 98–104
47. Mika S, Scholkopf B, Smola AJ, Muller KR, Scholz M, Ratsch G (1999) Kernel PCA and de-noising in feature spaces. In: Kearns MS, Solla SA, Cohn DA (eds) Advances in neural information processing systems, vol 11, pp 536–542
48. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12:2385–2404
49. Roth V, Steinhage V (1999) Nonlinear discriminant analysis using kernel functions. In: Proceedings of neural information processing systems, Denver, Nov 1999
50. Wang T-S, Zheng N-N, Li Y, Xu Y-Q, Shum H-Y (2003) Learning kernel-based HMMs for dynamic sequence synthesis. Graph Models 65:206–221
51. Zhang B-L, Zhang H, Sam Ge S (2004) Face recognition by applying wavelet subband representation and kernel associative memory. IEEE Trans Neural Networks 15(1):166–177
52. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Network 12(6):783–789
53. Peng J, Heisterkamp DR, Dai HK (2004) Adaptive quasiconformal kernel nearest neighbor classification. IEEE Trans Pattern Anal Mach Intell 26(5):656–661
54. Jiao L, Li Q (2006) Kernel matching pursuit classifier ensemble. Pattern Recogn 39:587–594

55. Zhang D, Chen S, Zhou Z-H (2006) Learning the kernel parameters in kernel minimum distance classifier. Pattern Recogn 39:133–135
56. Jiao L, Li Q (2006) Kernel matching pursuit classifier ensemble. Pattern Recogn 39:587–594
57. Ma B, Qu H-y, Wong H-s (2007) Kernel clustering-based discriminant analysis. Pattern Recogn 40(1):324–327
58. Szymkowiak Have A, Girolami MA, Larsen J (2006) Clustering via kernel decomposition. IEEE Trans Neural Networks 17(1):256–264
59. Kima D-W, Young Lee K, Lee D, Lee KH (2005) Evaluation of the performance of clustering algorithms in kernel-induced feature space. Pattern Recogn 38(4):607–611
60. Yang J, Yang J-y, Frangi AF (2003) Combined Fisherfaces framework. Image Vis Comput 21:1037–1044
61. Liu C (2004) Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Trans Pattern Anal Mach Intell 26(5):572–581
62. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Networks 14(1):117–226
63. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Networks 14(1):117–126
64. Liang Z, Shi P (2005) Kernel direct discriminant analysis and its theoretical foundation. Pattern Recogn 38:445–447
65. Liang Y, Li C, Gong W, Pan Y (2007) Uncorrelated linear discriminant analysis based on weighted pairwise Fisher criterion. Pattern Recogn 40:3606–3615
66. Liang Z, Shi P (2005) Uncorrelated discriminant vectors using a kernel method. Pattern Recogn 38:307–310
67. Yang J, Frangi AF, Yang J-y, Zhang D, Jin Z (2005) KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27(2):230–244
68. Belhumeur V, Hespanda J, Kiregeman D (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans PAMI 19:711–720
69. Yang J, Jin Z, Yang J-y, Zhang D, Frangi AF (2004) Essence of kernel Fisher discriminant: KPCA plus LDA. Pattern Recogn 37:2097–2100
70. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
71. Zheng W, Zou C, Zhao L (2005) Weighted maximum margin discriminant analysis with kernels. Neurocomputing 67:357–362
72. Wu X-H, Zhou J-J (2006) Fuzzy discriminant analysis with kernel methods. Pattern Recogn 39(11):2236–2239
73. Tao D, Tang X, Li X, Rui Y (2006) Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. IEEE Trans Multimedia 8(4):716–727
74. Huang J, Yuen PC, Chen W-S, Lai JH (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. In: Proceedings of the 6th IEEE international conference on automatic face and gesture recognition, pp 1352–1355
75. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst Man Cybern B Cybern 35(3):556–562
76. Chen W-S, Yuen PC, Huang J, Dai D-Q (2005) Kernel machine-based one-parameter regularized Fisher discriminant method for face recognition. IEEE Trans Syst Man Cybern B Cybern 35(4):658–669
77. Liang Z, Shi P (2004) Efficient algorithm for kernel discriminant analysis. Pattern Recogn 37(2):381–384
78. Liang Z, Shi P (2004) An efficient and effective method to solve kernel Fisher discriminant analysis. Neurocomputing 61:485–493

79. Yang MH (2002) Kernel Eigenfaces vs. Kernel Fisherfaces: face recognition using kernel methods. In: Proceedings of fifth IEEE international conference on automatic face and gesture recognition, pp 215–220
80. Zheng Y-j, Yang J, Yang J-y, Wu X-j (2006) A reformative kernel Fisher discriminant algorithm and its application to face recognition. Neurocomputing 69(13):1806–1810
81. Xu Y, Zhang D, Jin Z, Li M, Yang J-Y (2006) A fast kernel-based nonlinear discriminant analysis for multi-class problems. Pattern Recogn 39(6):1026–1033
82. Saadi K, Talbot NLC, Cawley GC (2007) Optimally regularised kernel Fisher discriminant classification. Neural Networks 20(7):832–841
83. Yeung D-Y, Chang H, Dai G (2007) Learning the kernel matrix by maximizing a KFD-based class separability criterion. Pattern Recogn 40(7):2021–2028
84. Shen LL, Bai L, Fairhurst M (2007) Gabor wavelets and general discriminant analysis for face identification and verification. Image Vis Comput 25(5):553–563
85. Ma B, Qu H-y, Wong H-s (2007) Kernel clustering-based discriminant analysis. Pattern Recogn 40(1):324–327
86. Liu Q, Lu H, Ma S (2004) Improving kernel Fisher discriminant analysis for face recognition. IEEE Trans Pattern Anal Mach Intell 14(1):42–49
87. Wang T-S, Zheng N-N, Li Y, Xu Y-Q, Shum H-Y (2003) Learning kernel-based HMMs for dynamic sequence synthesis. Graph Models 65:206–221
88. Yang J, Gao X, Zhang D, Yang J-y (2005) Kernel ICA: an alternative formulation and its application to face recognition. Pattern Recogn 38:1784–1787
89. Chang C, Ansari R (2005) Kernel particle filter for visual tracking. IEEE Signal Process Lett 12(3):242–245
90. Zhang P, Peng J, Domeniconi C (2005) Kernel pooled local subspaces for classification. IEEE Trans Syst Man Cybern 35(3):489–542
91. Zhang B-L, Zhang H, Sam Ge S (2004) Face recognition by applying wavelet subband representation and kernel associative memory. IEEE Trans Neural Networks 15(1):166–177
92. Zhu Z, He H, Starzyk JA, Tseng C (2007) Self-organizing learning array and its application to economic and financial problems. Inf Sci 177(5):1180–1192
93. Mulier F, Cherkassky V (1995) Self-organization as an iterative kernel smoothing process. Neural Comput 7:1165–1177
94. Ritter H, Martinetz T, Schulten K (1992) Neural computation and self-organizing maps: an introduction. Addison-Wesley, Reading
95. Zhang H, Zhang B, Huang W, Tian Q (2005) Gabor wavelet associative memory for face recognition. IEEE Trans Neural Networks 16(1):275–278
96. Sussner P (2003) Associative morphological memories based on variations of the kernel and dual kernel methods. Neural Networks 16:625–632
97. Wang M, Chen S (2005) Enhanced FMAM based on empirical kernel map. IEEE Trans Neural Networks 16(3):557–564
98. LeCun Y, Jackel LD, Bottou L, Brunot A, Corts C, Denker JS, Drucker H, Guyon I, Muller UA, Sackinger E, Simard P, Vapnik V (1995) Comparison of learning algorithms for handwritten digit recognition. In: Proceedings of international conferences on artificial neural networks, vol 2, pp 53–60
99. Scholkopf B (1997) Support vector learning. Oldenbourg-Verlag, Munich
100. Yang MH (2002) Kernel eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods. In: Proceedings of the fifth international conferences on automatic face and gesture recognition, pp 1425–1430
101. Kim KI, Jung K, Kim HJ (2002) Face recognition using kernel principal component analysis. IEEE Signal Process Lett 9(2):40–42
102. Pang S, Kim D, Bang SY (2003) Membership authentication in the dynamic group by face classification using SVM ensemble. Pattern Recogn Lett 24:215–225
103. Joachims T (1998) Text categorization with support vector machines. In: Proceedings of European conferences on machine learning, pp 789–794

104. Leopold E, Kindermann J (2002) Text categorization with support vector machines. How to represent texts in input space? Machine Learning 46:423–444
105. Pearson WR, Wood T, Zhang Z, Miller W (1997) Comparison of DNA sequences with protein sequences. Genomics 46(1):24–36
106. Hua S, Sun Z (2001) Support vector machine approach for protein subcellular localization prediction. Bioinformatics 17(8):721–728
107. Hua S, Sun Z (2001) A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J Mol Biol 308:397–407
108. Fyfe C, Corchado J (2002) A comparison of kernel methods for instantiating case based reasoning systems. Adv Eng Inform 16:165–178
109. Heisterkamp DR, Peng J, Dai HK (2001) Adaptive quasiconformal kernel metric for image retrieval. In: Proceedings of CVPR (2), pp 388–393

Chapter 2
Statistical Learning-Based Face Recognition

2.1 Introduction

Face recognition has the wide research and applications on many areas. Many surveys of face recognition are implemented. Different from previous surveys on from a single viewpoint of application, method, or condition, this book has a comprehensive survey on face recognition from practical applications, sensory inputs, methods, and application conditions. In the sensory inputs, we review face recognition from image-based, video-based, 3D-based, and hyperspectral image-based face recognition, and a comprehensive survey of face recognition methods from the viewpoints of signal processing and machine learning is implemented, such as kernel learning, manifold learning method. Moreover, we discuss the single-training-sample-based face recognition and under the variable poses. The prominent algorithms are described and critically analyzed, and relevant issues such as data collection, the influence of the small sample size, and system evaluation are discussed.

Statistical learning-based face recognition is a hot and popular research topic in recent years. As a subfield of pattern recognition, face recognition (or face classification) has become a hot research point. In pattern recognition and in image processing, feature extraction based on dimensionality reduction plays the important role in the relative areas. Feature extraction simplifies the amount of resources required to describe a large set of data accurately for classification and clustering. On the algorithms, when the input data are too large to be processed and it is suspected to be notoriously redundant (much data, but not much information), then the input data will be transformed into a reduced representation set of features with linear transformation or the nonlinear transformation. Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen, it is expected that the feature set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full-size input data.

Face recognition has been a popular research topic in the computer vision, image processing, and pattern recognition areas. Recognition performance of the practical face recognition system is largely influenced by the variations in

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_2,

illumination conditions, viewing directions or poses, facial expression, aging, and disguises. Face recognition provides the wide applications in commercial, law enforcement, and military, and so on, such as airport security and access control, building surveillance and monitoring, human–computer intelligent interaction and perceptual interfaces, smart environments at home, office, and cars. Many application areas of face recognition are developed based on two primary verification (one-to-one) and identification (one-to-many) tasks as shown in Table 2.1.

2.2 Face Recognition: Sensory Inputs

2.2.1 Image-Based Face Recognition

Image-based face recognition methods can be divided into feature-based and holistic methods. On feature-based face recognition, geometry-based face

Table 2.1 Face recognition applications

Areas	Tasks
Security [99, 10]	Access control to buildings
	Airports/seaports
	ATM machines
	Border checkpoints [100, 101]
	Computer/network security [43]
	Smart card [44]
Video indexing [102, 103]	Surveillance
	Labeling faces in video
Image database investigations [104]	Forensics
	Criminal justice systems
	Mug shot/booking systems
	Post-event analysis
	Licensed drivers' managing
	Benefit recipients
	Missing children
	Immigrants and police bookings
	Witness face reconstruction
General identity verification	Electoral registration
	Banking
	Electronic commerce
	Identifying newborns
	National IDs
	Passports
	Drivers' licenses
	Employee IDs
HCI [45, 105]	Ubiquitous aware
	Behavior monitoring
	Customer assessing

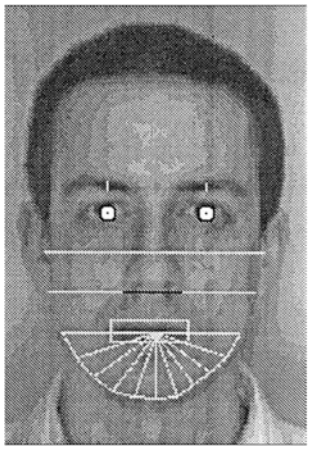


Fig. 2.1 Geometrical feature-based face recognition in [1]

recognition is the most popular method in the previous work. The work in [1] is a representative work, which computed a vector of 35 geometric features shown in Fig. 2.1, and a 90 % recognition rate was reported. But the high 100 % recognition accuracy is achieved by the same database with the experiments under the template-based face recognition. Other methods were proposed for geometry-based face recognition, including filtering and morphological operations [2], Hough transform methods [3], and deformable templates [4, 5]. Researchers applied 30-dimensional feature vector derived from 35 facial features as shown in Fig. 2.2 and reported a 95 % recognition accuracy on 685 images of database. These facial features are marked manually and had its limitations on autorecognition in the practical face recognition system. In the following research work [6], researchers presented an automatic feature extraction but less recognition accuracy.

On holistic methods, which attempt to identify faces using global representations, i.e., descriptions are based on the entire image rather than on local features of the face. Modular eigenfeature-based face recognition [7] deals with localized

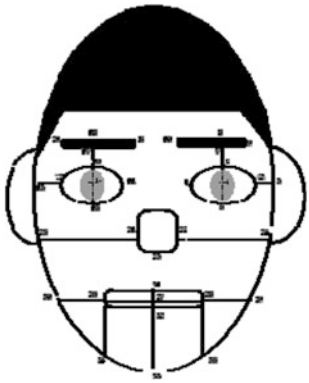


Fig. 2.2 Manually mark facial features [97]

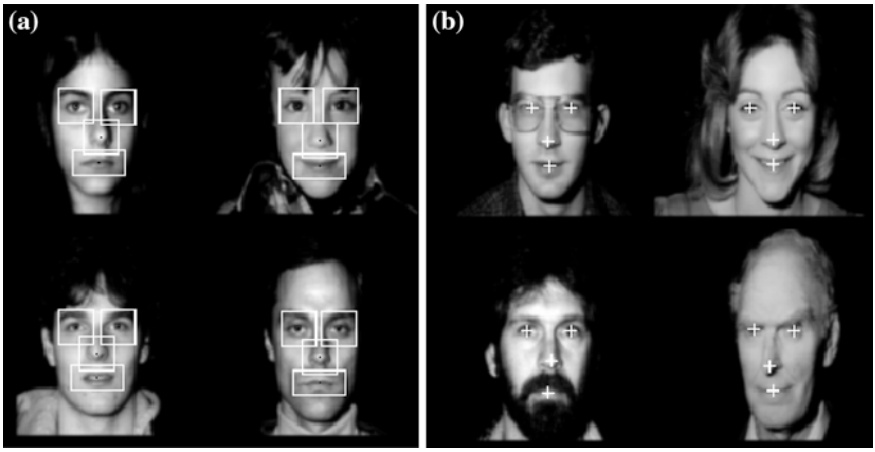


Fig. 2.3 **a** Examples of facial feature training templates used and **b** the resulting typical detections [7]

variations and a low-resolution description of the whole face in terms of the salient facial features as shown in Fig. 2.3.

As the famous face recognition method, principal component analysis (PCA) has been widely studied. Some recent advances in PCA-based algorithms include weighted modular PCA [8], adaptively weighted subpattern PCA [9], two-dimensional PCA [10, 11], multi-linear subspace analysis [12], eigenbands [13], symmetrical PCA [14].

2.2.2 Video-Based Face Recognition

With the development of video surveillance, video-based face recognition has widely used in many areas. Video-based face recognition system typically consists of face detection, tracking, and recognition [15]. In the practical video face recognition system, most of them applied a good frame to recognize a new face [16]. In [17], two types of image sequences were done in training and test procedure. As shown in Figs. 2.4 and 2.5, eight primary sequences were taken in a relatively constrained environment, and then a secondary sequence is recorded in unconstrained atmosphere (Fig. 2.6).

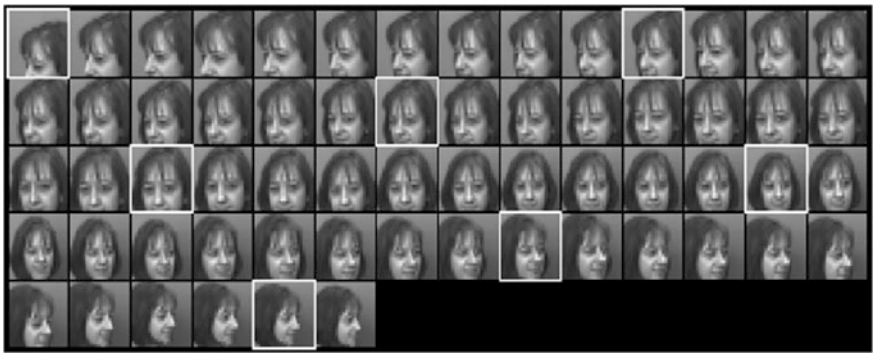


Fig. 2.4 A complete primary sequence for the class Carla [17]

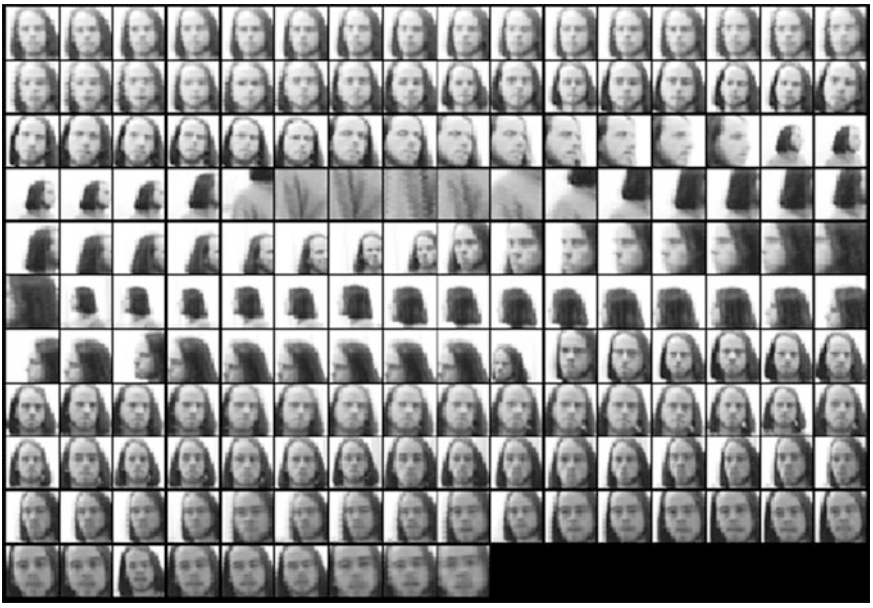


Fig. 2.5 A complete secondary sequence for the class Steve [17]

2.2.3 3D-Based Face Recognition

As shown in Fig. 2.7, sixteen examples with full-color frontal and profile view photographs are shown. The profile images were converted to grayscale images. To prevent participants from matching the faces by hairstyles and forehead fringes, the distances between the lowest hair cue in the forehead and the concave of the nose of all the faces were measured [18]. The minimum distance between the faces in the same set was taken as the standard length for all the faces in the same set,

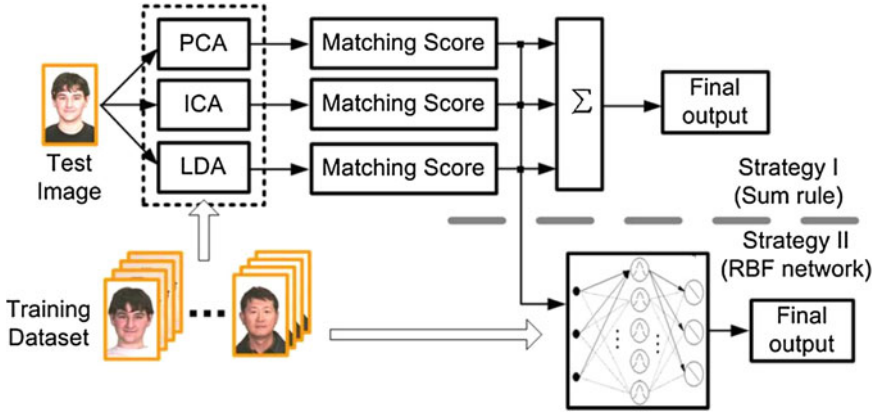


Fig. 2.6 Combining classifiers for face recognition [98]

Fig. 2.7 Examples of the front-view faces with their corresponding grayscale profiles [18]

and the faces in the same set were trimmed to the same extent based on the standard length.

2.2.4 Hyperspectral Image-Based Face Recognition

Multispectral and hyperspectral imaging with remote sensing purposes is widely used in environment reconnaissance, agriculture, forest, and mineral exploration. Multispectral and hyperspectral imaging obtains a set of spatially coregistered images with its spectrally contiguous wavelengths. Recently, it has been applied to biometrics, skin diagnosis, etc. Especially, some studies on hyperspectral face recognition have been reported very recently [19]. Researchers built an indoor hyperspectral face acquisition system shown in Fig. 2.8. For each individual, four sessions were collected at two different times (2 sessions each time) with an average time span of five months. The minimal interval is three months, and the maximum interval is ten months. Each session consists of three hyperspectral

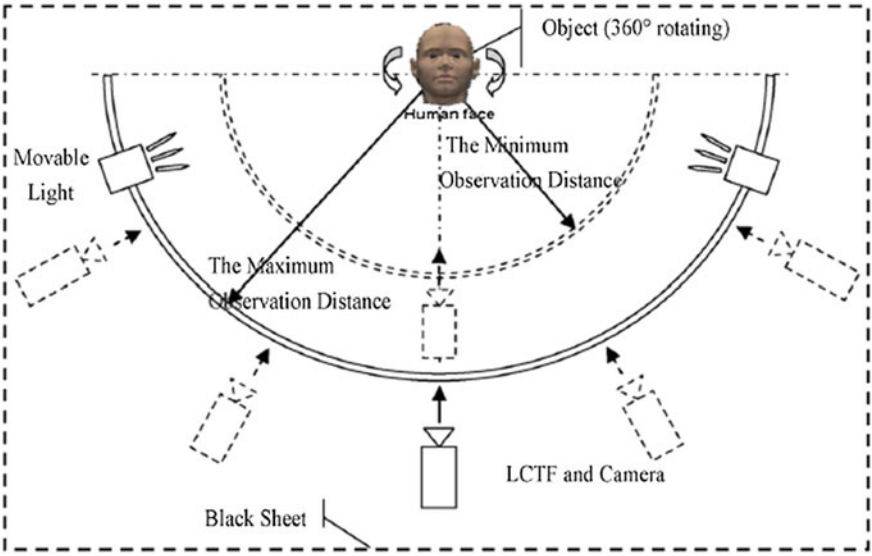


Fig. 2.8 Established hyperspectral face imaging system [19]

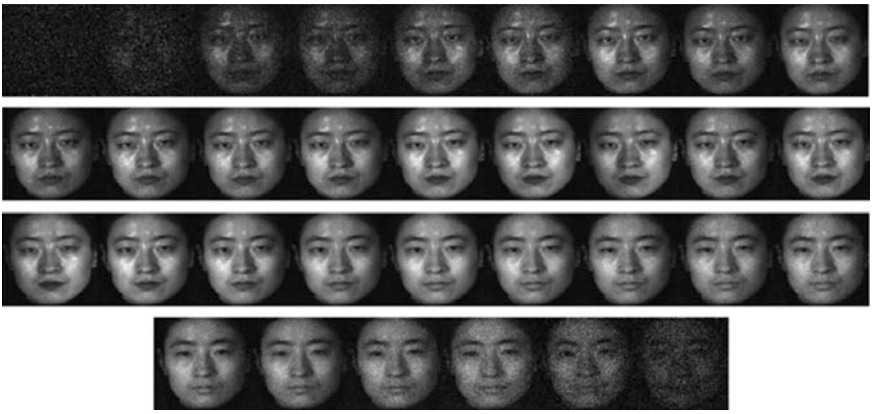


Fig. 2.9 Examples of a set of 33 bands of hyperspectral images from a person [19]

cubes—frontal, right, and left views with neutral expression. In the hyperspectral imaging system, the spectral range is from 400 to 720 nm with a step length of 10 nm with producing 33 bands in all. Some examples are shown in Fig. 2.9.

2.3 Face Recognition: Methods

2.3.1 Signal Processing-Based Face Recognition

An excellent face recognition method should consider what features are used to represent a face image and how to classify a new face image based on this representation. Current feature extraction methods can be classified into signal processing and statistical learning methods. On signal processing-based methods, feature extraction-based Gabor wavelets are widely used to represent the face image [20, 21], because the kernels of Gabor wavelets are similar to two-dimensional receptive field profiles of the mammalian cortical simple cells, which captures the properties of spatial localization, orientation selectivity, and spatial frequency selectivity to cope with the variations in illumination and facial expressions. On the statistical learning-based methods, the dimensionality reduction methods are widely used in the past works [22–27], and the PCA and LDA are widely used among the dimensionality reduction methods [28]. Recently, kernel-based nonlinear feature extraction methods were applied to face recognition [29–31], which has attracted much attention in the past research works [32, 33].

Recently, video-based technology has been developed and applied into many research topics including coding [34, 35], enhancing [36, 37], and face recognition as discussed in the previous section. In this section, Gabor-based face recognition technology is discussed. The use of Gabor filter sets for image segmentation has attracted quite some attention in the last decades. Such filter sets provide a promising alternative in view of the amount and diversity of "normal" texture features proposed in the literature. Another reason for exploiting this alternative is the outstanding performance of our visual system, which is known by now to apply such a local spectral decomposition. However, it should be emphasized that Gabor decomposition only represents the lowest level of processing in the visual system. It merely mimics the image coding from the input (cornea or retina) to the primary visual cortex (cortical hypercolumns), which, in turn, can be seen as the input stage for further and definitively more complex cortical processing. The nonorthogonality of the Gabor wavelets implies that there is redundant information in the filtered images (Fig. 2.10).

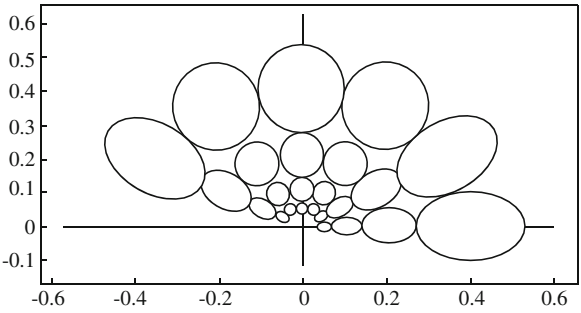


Fig. 2.10 The contours indicate the half-peak magnitude of the filter responses in the Gabor filter dictionary

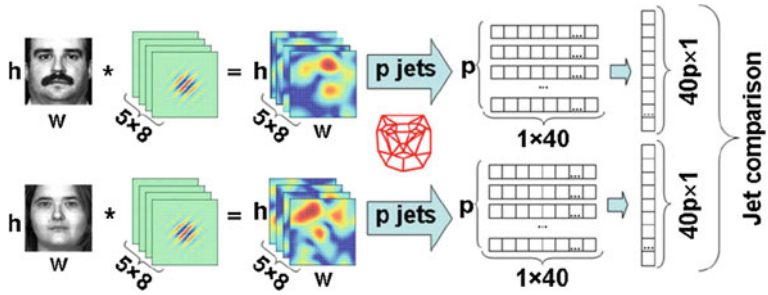


Fig. 2.11 Outline of analytical methods [38]

Current Gabor-based face recognition can be divided into two major types: analytical methods and holistic methods [38]. The flow of analytical method is shown in Fig. 2.11, and the one of the holistic methods is shown in Fig. 2.12. Based on how they select the nodes, analytical methods can be divided into graph-matching-based, manual detection (or other nongraph algorithms), and enhanced methods as shown in Table 2.2. As show in Fig. 2.12, Holistic methods consider Gabor convolutions as a whole and therefore usually rely on an adequate pre-processing, like face alignment, size normalization, and tilt correction. However, these methods still endure the dimensionality problem. So in the practical applications, dimensionality reduction methods such as PCA and LDA should be implemented to reduce the dimensionality of the vectors [39].

2.3.2 A Single Training Image per Person Algorithm

Face recognition has received more attention from the industrial communities in the recent years owing to its potential applications in information security, law enforcement and surveillance, smart cards, access control. In many practical applications, owing to the difficulties of collecting samples or storage space of

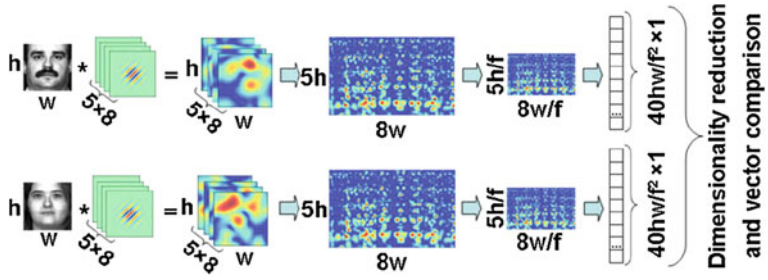


Fig. 2.12 Outline of holistic methods [38]

Table 2.2 Gabor-based face recognition [38]

Types	Major methods
Graph-based	EBGM
	DLA
Nongraph-based	Manual detection
	Ridge/valley detection
	Nonuniform sampling
	Gaussian mixture models
Enhanced	Optimal Gabor parameters
	Gabor + Adaboost
Downsampled Gabor+ PCA/LDA	GFC
	HEGFC
Downsampled Gabor Kernel PCA/LDA	Gabor kernel PCA
	Gabor + KDDA
Gabor 2D methods	Gabor + 2DPCA
	Gabor + B2DPCA
	Gabor + (2D)2PCA
Local binary patterns	LGBPHS
	GBC
	HGPP
No downsampling	Multichannel Gabor + PCA

systems, only one sample image per person is stored in the system, so the research of face recognition from one sample per person, owing to its own advantages (easy collecting of samples, less storage, and computational cost), has been a subresearch topic in the face recognition area. The traditional method such as Fisherface fails when each person just has one training face sample available because of nonexistence of the intraclass scatter. Recently, researchers have proposed many algorithms, such as (PC)2A [40], E(PC)2A [41], and SVD perturbation [42], for face recognition with one training image per person. But these algorithms still endure some problem. For example, the procedure of E(PC)2A is divided into two stages: (1) constructing a new image by combining the first-order and second-order projected images and the original image; (2) performing PCA on the newly combined training images. In the second stage, the combined image matrix should be mapped onto a 1D vector in advance in order to perform PCA. This causes the high storage and computational cost. In order to enhance the practicability of the face recognition system, we propose a novel algorithm so-called 2D(PC)2A for face recognition with one training image per person in this letter. 2D(PC)2A performs PCA on the set of combined training images directly without mapping the image matrix to 1D vector. Thus, 2D(PC)2A can directly extract feature matrix from the original image matrix. This leads to that much less time is required for training and feature extraction. Further, experiments implemented on two popular databases show that the recognition performance of 2D(PC)2A is better than that of classical E(PC)2A.

In the real-world application of face recognition system, owing to the difficulties of collecting samples or storage space of systems, only one sample image per person is stored in the system, which is so-called one sample per person problem. Moreover, pose and illumination have impact on recognition performance. In this letter, we propose a novel pose and illumination robust algorithm for face recognition with a single training image per person to solve the above limitations. Experimental results show that the proposed algorithm is an efficient and practical approach for face recognition.

The procedure of 2D(PC)2A can be divided into the three stages: (1) creating the combined image from the original image $I(m,\ n)$ with $M \times N$ pixels $(I(m,n) \in [0,1],\ m \in [1,M],\ n \in [1,N])$; (2) performing 2DPCA on the combined images; (3) classifying a new face based on assembled matrix distance (AMD). The detailed procedure is described as follows:

Step 1 Create the combined image. In order to effectively recognize faces with only one example image per class, we derive a combined image from the original image by the first-order and second-order projection. Firstly, the first-order projected image $P_1(m,n)$ and second-order projected image $P_2(m,n)$ are created as follows:

$$P_1(m,n) = \frac{V_1(m)H_1(n)}{MN\overline{I}} \tag{2.1}$$

$$P_2(m,n) = \frac{V_2(m)H_2(n)}{MN\overline{J}} \tag{2.2}$$

where $V_1(m) = \frac{1}{N}\sum_{p=1}^{N} I(m,p)$ and $H_1(n) = \frac{1}{M}\sum_{q=1}^{M} I(q,n)$, and $\overline{I}$ is the mean value of $I(m,n)$, and $V_2(m) = \frac{1}{N}\sum_{n=1}^{N} J(m,n)$ and $H_2(n) = \frac{1}{M}\sum_{m=1}^{M} J(m,n)$ and $J(m,n) = I(m,n)^2$, and $\overline{J}$ is the mean value of $J(m,n)$. Secondly, the combined image can be created as follows:

$$I_p(m,n) = \frac{I(m,n) + \alpha P_1(m,n) + \beta P_2(m,n)}{1+\alpha+\beta} \tag{2.3}$$

Step 2 Perform 2DPCA. Instead of performing PCA on the set of combined images, 2D(PC)2A performs two-dimensional PCA on the image matrix directly rather than 1D vectors for covariance matrix estimation, thus claimed to be more computationally cheap and more suitable for small sample size problem. Let the combined image I_{pj} $(j = 1, 2, \ldots, C)$ the average image of all training samples be $\overline{I_p}$, then the image covariance matrix S_T can be evaluated as follows:

$$S_T = \frac{1}{C}\sum_{j=1}^{C}\left(I_{\mathrm{pj}} - \overline{I_p}\right)^T\left(I_{\mathrm{pj}} - \overline{I_p}\right) \tag{2.4}$$

Then, a set of optimal projection axis of 2DPCA $\{w_1, w_2, \ldots, w_d\}$, which are then used for feature extraction, can be obtained by maximizing the image scatter criterion

$$J(W) = W^T S_T W \tag{2.5}$$

The low-dimensional feature matrix Y of a combined image matrix I_p can be obtained as follows:

$$Y = I_p W_{\mathrm{opt}} \tag{2.6}$$

where $W_{\mathrm{opt}} = \{w_1, w_2, \ldots, w_d\}$. In Eq. (2.6), the dimension of 2DPCA projector W_{opt} is $N \times d$, and the dimension of 2DPCA feature matrix Y is $M \times d$.

Step 3 Implement AMD for classification. After the feature matrices are extracted from the original images based on 2D(PC)2A in Step 1 and 2, the nearest neighbor criterion is applied to classification based on the distance between two feature matrices. Unlike E(PC)2A approach to produce a feature vector, 2D(PC)2A directly extracts a feature matrix from an original image matrix. So, we apply AMD metric to calculate the distance between two feature matrices. Given two feature matrices $A = (a_{\mathrm{ij}})_{M\times d}$ and $B = (b_{\mathrm{ij}})_{M\times d}$, the AMD $d_{\mathrm{AMD}}(A, B)$ is obtained as follows:

$$d_{\mathrm{AMD}}(A,B) = \left(\sum_{j=1}^{d}\left(\sum_{i=1}^{M}\left(a_{\mathrm{ij}} - b_{\mathrm{ij}}\right)^2\right)^{(1/2)p}\right)^{1/p} \tag{2.7}$$

After calculating the AMD between the feature matrix of the test sample and the training sample, we apply nearest neighbor criterion to classification based on AMD.

In this section, we implement experiments on ORL [43], YALE [44], and UMIST databases [45] to evaluate the proposed algorithm. Firstly, we introduce the face databases as follows.

Olivetti Research Laboratory (ORL) face database, developed at the ORL, Cambridge, UK, is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations in the images are across pose, time, and facial expression. Some image examples are shown in Fig. 2.1.

YALE face database was constructed at the YALE Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting conditions (left-light, center-light, and

right-light), and different facial expressions (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Some image examples are shown in Fig. 2.2.

UMIST face database consists of 564 images of 20 people. Each covers a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. Each subject exists in their own directory labeled 1a, b, … t, and images are numbered sequentially as they were taken. Some image examples are shown in Fig. 2.3.

We implement experiments on ORL and YALE with two manners. A deterministic manner: The training set and the testing set are constructed as shown in Tables 2.1 and 2.2. Here, our goal is to have a good look at the performance of specific partition of the database; thus, we can see how much the influence of recognition rate under the different pose, illumination, and expression (PIE). A random manner: From ORL face database, we randomly select one image from each subject, and the rest images are used to test the performance. Only one image of each person randomly selected from YALE database is used to construct the training set, and the rest images of each person are used to test the performance of the algorithms. Moreover, we implement experiments on UMIST face database on a random manner.

It is worthy to emphasize the following points. (1) We run experiments for 10 times, and the average rate is used to evaluate the classification performance. (2) The experiments are implemented on a Pentium 3.0 GHz computer with 512-MB RAM and programmed in the MATLAB platform (version 6.5). (3) To reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels. Similarly, the images from YALE databases are cropped to the size of 100×100 pixels, and finally, a subimage procedure crops the face image to the size of 112×92 to extract the facial region on UMIST face database.

We also implement other popular methods such as PCA, (PC)2A, E(PC)2A, and SVD perturbation for face recognition with single training sample per person. In our experiments, we select $\alpha = 0.125$ and $\beta = 0.05$ for 2D(PC)2A and E(PC)2A. As results shown in Tables 2.3, 2.4, 2.5, 2.6, and 2.7, the proposed algorithm gives a highest recognition rate compared with other popular methods. Moreover, since 2D(PC)2A deals with matrix directly instead of mapping onto 1D vector as E(PC)2A or (PC)2A, it is apparent that 2D(PC)2A is more efficient than E(PC)2A or (PC)2A. So, we say that 2D(PC)2A method is an efficient and practical approach for face recognition (Tables 2.8, 2.9).

Table 2.3 Deterministic training and test set on ORL face database

	Training set	Test set
ORL_A	1#	2#, 3#, 4#, 5#, 6#, 7#, 8#, 9#, 10#
ORL_B	2#	1#, 3#, 4#, 5#, 6#, 7#, 8#, 9#, 10#
ORL_C	3#	1#, 2#, 4#, 5#, 6#, 7#, 8#, 9#, 10#
ORL_D	4#	1#, 2#, 3#, 5#, 6#, 7#, 8#, 9#, 10#
ORL_E	5#	1#, 2#, 3#, 4#, 6#, 7#, 8#, 9#, 10#

Notes 1# denotes the first image of each person, 2# denotes the second image of each person, and other images are marked with the same ways

Table 2.4 Deterministic training and test set on YALE face database

	Training set	Test set
YALE_a	1#	2#, 3#, 4#, 5#, 6#, 7#, 8#, 9#, 10#, 11#
YALE _b	2#	1#, 3#, 4#, 5#, 6#, 7#, 8#, 9#, 10#, 11#
YALE _c	3#	1#, 2#, 4#, 5#, 6#, 7#, 8#, 9#, 10#, 11#
YALE _d	4#	1#, 2#, 3#, 5#, 6#, 7#, 8#, 9#, 10#, 11#
YALE _e	5#	1#, 2#, 3#, 4#, 6#, 7#, 8#, 9#, 10#, 11#

Notes 1# denotes the first image of each person, 2# denotes the second image of each person, and other images are marked with the same ways

Table 2.5 Recognition performance on ORL database in random manner

Algorithms	PCA	2DPCA	(PC)2A	E(PC)2A	SVD	2D(PC)2A
Recognition rate	0.54	0.54	0.56	0.57	0.55	0.60

Table 2.6 Recognition performance on YALE database in random manner

Algorithms	PCA	2DPCA	(PC)2A	E(PC)2A	SVD	2D(PC)2A
Recognition rate	0.54	0.56	0.55	0.56	0.54	0.61

Table 2.7 Recognition performance on ORL database in deterministic manner

Algorithms	PCA	2DPCA	(PC)2A	E(PC)2A	SVD	2D(PC)2A
ORL_A	0.55	0.54	0.57	0.58	0.56	0.61
ORL_B	0.55	0.55	0.58	0.59	0.59	0.62
ORL_C	0.55	0.56	0.57	0.58	0.59	0.61
ORL_D	0.57	0.56	0.59	0.60	0.59	0.63
ORL_E	0.57	0.57	0.61	0.62	0.60	0.64

Table 2.8 Recognition performance on YALE database in deterministic manner

Algorithms	PCA	2DPCA	(PC)2A	E(PC)2A	SVD	2D(PC)2A
YALE_a	0.55	0.56	0.55	0.57	0.55	0.62
YALE _b	0.56	0.57	0.57	0.58	0.56	0.63
YALE _c	0.54	0.55	0.54	0.56	0.55	0.61
YALE _d	0.57	0.58	0.57	0.59	0.58	0.64
YALE _e	0.56	0.59	0.58	0.58	0.58	0.63

Table 2.9 Recognition performance on UMIST face database in random manner

Algorithms	PCA	2DPCA	(PC)2A	E(PC)2A	SVD	2D(PC)2A
Recognition rate	0.56	0.57	0.59	0.60	0.59	0.65

Although the proposed algorithm gives a highest recognition rate compared with other popular methods, the highest recognition rate is still not so high (only about 0.60) owing to PIE problem of face recognition. So in the future research work, we will pay attention to solve the PIE problem to enhance the whole recognition rate of the algorithm. Some essential questions are included here under to be answered in the future: (1) Are there other methods of choosing α and β in the practical application? (2) In the experiments, the parameter p for AMD is chosen with the experiments. Is there other alternative method to choose this parameter? (3) 2D(PC)2A gives higher recognition accuracy, but the recognition rate is not so high. How to increase the recognition performance of the algorithm is a key problem in the future work.

2.4 Statistical Learning-Based Face Recognition

Learning-based method aims to reduce the dimensionality of the original data with mapping-based dimensionality reduction. The learning methods can be divided into two kinds of linear and nonlinear methods. The linear projection-based methods are used to find an optimal matrix projection to achieve the separable class discriminant of the data in the feature space. As shown in Fig. 2.13, the data in the original space are projected onto feature space, where the excellent class discriminative ability is achieved. This method is widely used in many applications, and the excellent results are reported in the many areas. But in some applications, such as face recognition, different face images have the similar pixel distributions, contour geometries, which cause more difficulties on recognition. Since the influences of poses, illuminations, and expressions, the different face images from a same person have the complex changes, and these changes cannot

Fig. 2.13 Some examples from ORL face database (the images come from the ORL, Cambridge, UK)

Fig. 2.14 Some examples from YALE face database (the images come from the YALE Center for Computational Vision and Control)

Fig. 2.15 Some examples from UMIST face database (the images come from University of Manchester Institute of Science and Technology)

be described with the linear projection. The previous linear methods perform not well on this case. It is not able to find the optimal projection for the good class discriminative ability of data in the feature space as shown in Fig. 2.14. The basic solution of this problem is shown in Figs. 2.15, 2.16, and 2.17.

2.4.1 Manifold Learning-Based Face Recognition

Feature extraction with dimensionality reduction is an important step and essential process in embedding data analysis [46]. Linear dimensionality reduction aims to develop a meaningful low-dimensional subspace in a high-dimensional input space such as PCA and LDA. LDA is to find the optimal projection matrix with Fisher criterion through considering the class labels, and PCA seeks to minimize the mean square error criterion. PCA is generalized to form the nonlinear curves such as principal curves [47] or principal surfaces [48]. Principal curves and surfaces

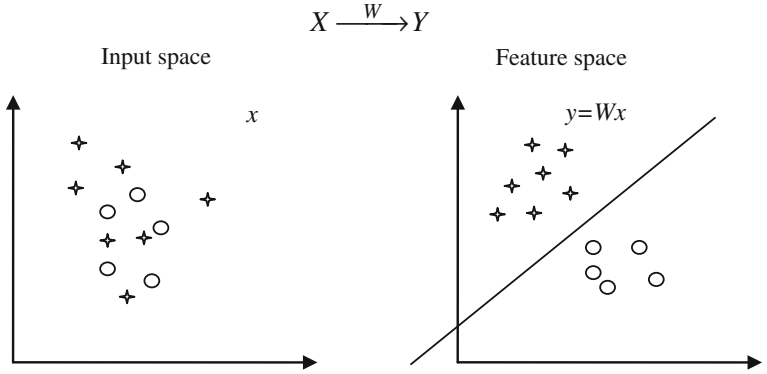


Fig. 2.16 Linear separable problem

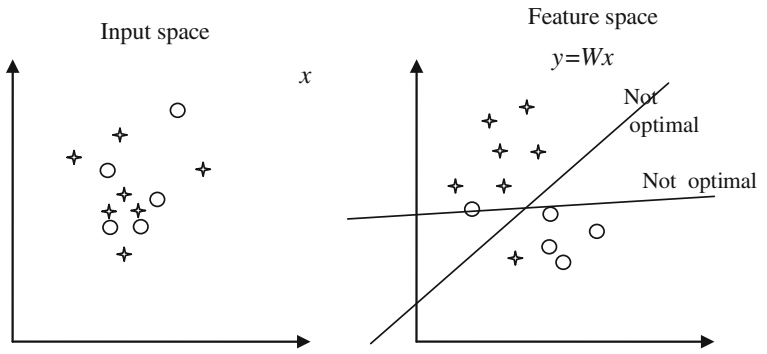


Fig. 2.17 Linear inseparable problem

are nonlinear generalizations of principal components and subspaces, respectively. The principal curves are essentially equivalent to self-organizing maps (SOM) [49]. With the extended SOM, ViSOM preserves directly the distance information on the map along with the topology [50], which represents the nonlinear data [51] and represents a discrete principal curve or surface through producing a smooth and graded mesh in the data space. Recently, researchers proposed other manifold algorithms such as Isomap [52], locally linear embedding (LLE) [53], and locality preserving projection (LPP) [54]. LPP projects easily any new data point in the reduced representation space through preserving the local structure and intrinsic geometry of the data space [55]. Many improved LPP algorithms were proposed in recent years. Zheng et al. used the class labels of data points to enhance its discriminant power in the low-dimensional mapping space to propose supervised LPP (SLPP) for face recognition [56]. However, LPP is not orthogonal, which makes it difficult to reconstruct the data, so researchers applied the class information to present orthogonal discriminant locality preserving projections

(ODLPP) for face recognition through orthogonalizing the basis vectors of the face subspace [57]. Cai et al. proposed the OLPP to produce orthogonal basis functions with more power of preserving locality than LPP [58]. OLPP was reported to have more discriminating power than LPP. Yu et al. introduced a simple uncorrelated constraint into the objective function to present uncorrelated discriminant locality preserving projections (UDLPP) with the aim of preserving the within-class geometric structure but maximizing the between-class distance [59]. In order to improve the performance of LPP on the nonlinear feature extraction, researchers perform UDLPP in reproducing kernel Hilbert space to develop Kernel UDLPP for face recognition and radar target recognition. Feng et al. presented an alternative formulation of Kernel LPP (KLPP) to develop a framework of KPCA + LPP algorithm [60]. In recent research, locality preserving projection and its improved methods are used in many areas, such as object recognition [61, 62], face detection [63, 64], and image analysis [65]. For any special image-based applications, such as face recognition, researchers proposed 2D LPP which extracts directly the proper features from image matrices without transforming one matrix into one vector [66, 67]. Both PCA and LPP are unsupervised learning methods, and LDA is supervised learning method. One of the differences between PCA and LPP lies in the global or local preserving property, that is, PCA seeks to preserve the global property, while LPP preserves the local structure. The locality preserving property leads to the fact that LPP outperforms PCA. Also, as the global method, LDA utilizes the class information to enhance its discriminant ability which causes LDA to outperform PCA on classification. But the objective function of LPP is to minimize the local quantity, i.e., the local scatter of the projected data. This criterion cannot be guaranteed to yield a good projection for classification purposes. So, it is reasonable to enhance LPP on classification using the class information like LDA.

2.4.2 Kernel Learning-Based Face Recognition

Some algorithms using the kernel trick are developed in recent years, such as kernel principal component analysis (KPCA), kernel discriminant analysis (KDA), and support vector machine (SVM). KPCA was originally developed by Scholkopf et al. in 1998, while KDA was firstly proposed by Mika et al. in 1999. KDA has been applied in many real-world applications owing to its excellent performance on feature extraction. Researchers have developed a series of KDA algorithms (Juwei Lu [68], Baudat and Anouar [69], Liang and Shi [70–71], Yang [72, 73], Lu [74], Zheng [75], Huang [76], Wang [77] and Chen [78], Liang [79], Zheng [80], Tao [81], Xu [82], Saadi [83], Yeung [84], Shen [85], Ma [86], Wu [87], Liu [88]). Because the geometrical structure of the data in the kernel mapping space, which is totally determined by the kernel function, has significant impact on the performance of these KDA methods, the separability of the data in the feature space could be even worse if an inappropriate kernel is used. In order to improve

the performance of KDA, many methods of optimizing the kernel parameters of the kernel function are developed in recent years (Huang [76], Wang [77] and Chen [78]). However, choosing the parameters for kernel just from a set of discrete values of the parameters does not change the geometrical structures of the data in the kernel mapping space. In order to overcome the limitation of the conventional KDA, we introduce a novel kernel named quasi-conformal kernel which were widely studied in the previous work [89, 90], where the geometrical structure of data in the feature space is changeable with the different parameters of the quasi-conformal kernel. The optimal parameters are computed through optimizing an objective function designed with the criterion of maximizing the class discrimination of the data in the kernel mapping space.

2.5 Face Recognition: Application Conditions

Face recognition has its limitations in practical applications including poses and training samples collection. As shown in Table 2.3, the current methods are divided into the following types, including pose transformation in image space, pose transformation in feature space, general algorithms, generic shape-based methods, feature-based 3D reconstruction, 2D techniques for face recognition across pose, and local approaches (Table 2.10).

The performance of face recognition system is influenced by many factors. And the limited number of training samples per person is a major factor. Now, it is still a question whether it deserves further investigation. Firstly, the extreme case of one sample per person really commonly happens in real scenarios and this problem needs be carefully addressed. Secondly, storing only one sample per person in the database has several advantages desired by most real-world applications. In fact, the practical face recognition system with only single training sample per person has its advantage owing to the following factors of easy sample collection, storage cost saving, and computational cost saving. Current algorithms can be divided into three types including holistic methods, local methods, and hybrid methods [91, 92] (Table 2.11).

Holistic methods: These methods identify a face using the whole face image as input. The main challenge faced by these methods is how to address the extremely small sample problem. Local methods: These methods use the local facial features for recognition. Care should be taken when deciding how to incorporate global configurational information into local face model. Hybrid methods: These methods use both the local and holistic features to recognize a face. These methods have the potential to offer better performance than individual holistic or local methods, since more comprehensive information could be utilized. Table 2.4 summarizes algorithms and representative works for face recognition from a single image. Below, we discuss the motivation and general approach of each category first, and then, we give the review of each method, discussing its advantages and disadvantages.

Table 2.10 Face recognition methods across pose

Main ideas	Methods
Pose transformation in image space	Parallel deformation [106]
	Pose parameter manipulation [107]
	Active appearance models [108, 109]
	Linear shape model [110]
	Eigen light-field [111]
Pose transformation in feature space	Kernel methods (kernelPCA [112, 113]
	Kernel FDA [114, 115])
	Expert fusion [116]
	Correlation filters [117]
	Local linear regression [118]
	Tied factor analysis [119]
General algorithms	Principal component analysis [120–122]
	Artificial neural network (convolutional networks [123])
	Line edgemaps [124]
	Directional corner point [125]
Generic shape-based methods	Cylindrical 3D pose recovery [126]
	Probabilistic geometry assisted face recognition [127]
	Automatic texture synthesis [128]
Feature-based 3D reconstruction	Composite deformable model [129]
	Jiang's method [130]
	Multi-level quadratic variation minimization [131]
	Image-based 3D reconstruction
	Morphable model [132, 133]
	Illumination cone model [134, 135]
	Stereo matching [136]
2D techniques for face recognition across pose	Real view-based matching
	Beymer's method [137]
	Panoramic view [138]
Local approaches	Template matching [139]
	Modular PCA [140]
	Elastic bunch graph matching [141]
	Local binary patterns [142]

In many practical applications, owing to the difficulties of collecting samples or storage space of systems, only one sample image per person is stored in the system, so the research of face recognition from one sample per person, owing to its own advantages (easy collecting of samples, less storage, and computational cost), has been a subresearch topic in the face recognition area. The traditional method such as Fisherface [93] fails when each person just has one training face sample available because of nonexistence of the intraclass scatter. Recently, researchers have proposed many algorithms, such as (PC)2A [94], as shown in Fig. 2.18, and these two projections reflect the distribution of the salient facial

Table 2.11 Current face recognition methods from a single training sample

Main ideas	Methods
Local feature-based	Graph matching methods [143–146]
	Use directional corner points (DCP) features for recognition [147]
Local appearance-based	Modified LDA method [148, 149]
	SOM learning-based recognition [150, 151]
	HMM method [152]
	Local probabilistic subspace method [153]
	Fractal-based face recognition [154]
	Hybrid local features [155]
	Local probabilistic subspace method [153]
	Face recognition with local binary patterns [156]
Extensions of principal component analysis (PCA)	Use noise model to synthesize new face [157]
	Enrich face image with its projections [158]
	Select discriminant eigenfaces for face recognition [159]
	Two-dimensional PCA [160]
Enlarge the size of training set	ROCA [161], imprecisely location method [153], E(PC)2A [162]
	View synthesis using prior class-specific information [163]

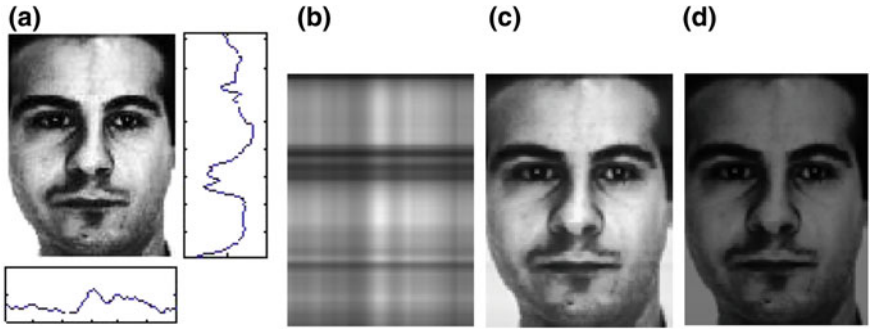


Fig. 2.18 Some sample images in (PC)2A method. **a** Original face image and its horizontal and vertical profiles. **b** First-ordered projection map. **c** First-ordered projection-combined image. **d** Second-ordered combined image

features that are useful for face recognition. Other enhanced algorithms are E(PC)2A [95] and SVD perturbation [96], for face recognition with one training image per person. But these algorithms still endure some problem. For example, the procedure of E(PC)2A is divided into two stages: (1) constructing a new image by combining the first-order and second-order projected images and the original image; (2) performing PCA on the newly combined training images. In the second stage, the combined image matrix should be mapped onto a 1D vector in advance in order to perform PCA. This causes the high storage and computational cost.

In order to enhance the practicability of the face recognition system, we propose a novel algorithm so-called 2D(PC)2A for face recognition with one training image per person in this letter. 2D(PC)2A performs PCA on the set of combined training images directly without mapping the image matrix to 1D vector. Thus, 2D(PC)2A can directly extract feature matrix from the original image matrix. This leads to that much less time is required for training and feature extraction. Further, experiments implemented on two popular databases show that the recognition performance of 2D(PC)2A is better than that of classical E(PC)2A (Fig. 2.18).

References

1. Brunelli R, Poggio T (1993) Face recognition: features versus templates. IEEE Trans Pattern Anal Mach Intell 15:1042–1052
2. Graf HP, Chen T, Petajan E, Cosatto E (1995) Locating faces and facial parts. In: Proceedings of international workshop on automatic face- and gesture-recognition, pp 41–46
3. Nixon M (1985) Eye spacing measurement for facial recognition. In: Proceedings of SPIE, pp 279–285
4. Roeder N, Li X (1995) Experiments in analyzing the accuracy of facial feature detection. In: Proceedings of vision interface'95, pp 8–16
5. Colombo C, Bimbo AD, Magistris SD (1995) Human-computer interaction based on eye movement tracking. Comput Architect Mach Percept, pp 258–263
6. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural network approach. In: Proceedings of IEEE transactions on neural networks, special issue on neural networks and pattern recognition, pp 1–24
7. Pentland A, Moghaddam B, Starner T (1994) Viewbased and modular eigenspaces for face recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 84–90
8. Kumar AP, Das S, Kamakoti V (2004) Face recognition using weighted modular principle component analysis. In: Proceedings of neural information processing, vol 3316. Lecture notes in computer science, Springer, Berlin/Heidelberg, pp 362–367
9. Tan KR, Chen SC (2005) Adaptively weighted subpattern PCA for face recognition. Neurocomputing 64:505–511
10. CNN (2003) Education school face scanner to search for sex offenders. The Associated Press, Phoenix, Arizona
11. Meng J, Zhang W (2007) Volume measure in 2DPCA based face recognition. Pattern Recogn Lett 28:1203–1208
12. Vasilescu MAO, Terzopoulos D (2003) Multilinear subspace analysis of image ensembles. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 93–99
13. Cavalcanti GDC, Filho ECBC (2003) Eigenbands fusion for frontal face recognition. In: Proceedings of IEEE international conference on image processing, vol 1, pp 665–668
14. Yang Q, Ding XQ (2003) Symmetrical principal component analysis and its application in face recognition. Chin J Comput 26:1146–1151
15. Torres L, Lorente L, Vilà J (2000) Face recognition using self-eigenfaces. In: Proceedings of international symposium on image/video communications over fixed and mobile networks. Rabat, Morocco, pp 44–47
16. Chellappa R, Wilson CL, Sirohey S (1995) Human and machine recognition of faces: a survey. Proc IEEE 83:705–740

17. Howell A, Buxton H () Towards unconstrained face recognition from image sequences. In: Proceedings of the second IEEE international conference on automatic face and gesture recognition, pp 224–229
18. Schwaninger A, Yang J (2011) The application of 3D representations in face recognition. Vision Res 51:969–977
19. Di W, Zhang L, Zhang D, Pan Q (2010) Studies on hyperspectral face recognition in visible spectrum with feature band selection. IEEE Trans Syst Man Cybern Part A Syst Hum 40(6):1354–1361
20. Liu C, Wechsler H (2002) Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. IEEE Trans Image Process 11(4):467–476
21. Zhang H, Zhang B, Huang W, Tian Q (2005) Gabor wavelet associative memory for face recognition. IEEE Trans Neural Networks 16(1):275–278
22. Xie Y, Setia L, Burkhardt H (2008) Face image retrieval based on concentric circular fourier-Zernike descriptors. Int J Innovative Comput Inf Control 4(6):1433–1444
23. Ryu H, Dinh V, Kim M (2007) Real-time multi-view face tracking combining learning based classifier and template matching. ICIC Express Lett 1(2):185–189
24. Li JB, Pan JS, Chu SC (2008) Kernel class-wise locality preserving projection. Inf Sci 178(7):1825–1835
25. Pan JS, Li JB, Lu ZM (2008) Adaptive quasiconformal kernel discriminant analysis. Neurocomputing 71:2754–2760
26. Li JB, Chu SC, Pan JS, Ho JH (2007) Adaptive data-dependent matrix norm based gaussian kernel for facial feature extraction. Int J Innovative Comput Inf Control 3(5):1263–1272
27. Li JB, Pan JS (2008) A novel pose and illumination robust face recognition with a single training image per person algorithm. Chin Opt Lett 6(4):255–257
28. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
29. Ruiz A, López de Teruel PE (2001) Nonlinear kernel-based statistical pattern analysis. IEEE Trans Neural Networks 12:16–32
30. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 12:181–201
31. Liu Q, Lu H, Ma S (2004) Improving kernel fisher discriminant analysis for face recognition. IEEE Trans Pattern Anal Mach Intell 14(1):42–49
32. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 12:181–201
33. Sahbi H (2007) Kernel PCA for similarity invariant shape recognition. Neurocomputing 70:3034–3045
34. Wang Hui, Liang J, Jay Kuo C–C (2010) Overview of robust video streaming with network coding. J Inf Hiding Multimedia Sig Process 1(1):36–50
35. Lou J, Liu S, Vetro A, Sun M-T (2010) Trick-play optimization for H.264 video decoding. J Inf Hiding Multimedia Sig Process 1(2):132–144
36. Rao Y, Chen L (2012) A survey of video enhancement techniques. J Inf Hiding Multimedia Sig Process 3(1):71–99
37. Rao Y, Chen L (2011) An efficient contourlet-transform-based algorithm for video enhancement. J Inf Hiding Multimedia Sig Process 2(3):282–293
38. Serrano A, de Diego IM, Conde C, Cabello E (2010) Recent advances in face biometrics with Gabor wavelets: a review. Pattern Recogn Lett 31:372–381
39. Parviz M, Moin MS (2011) Boosting approach for score level fusion in multimodal biometrics based on AUC maximization. J Inf Hiding Multimedia Sig Process 2(1):51–59
40. Wu J, Zhou Z-H (2002) Face recognition with one training image per person. Pattern Recogn Lett 23(14):1711–1719
41. Chen SC, Zhang DQ, Zhou Z-H (2004) Enhanced (PC)2A for face recognition with one training image per person. Pattern Recogn Lett 25(10):1173–1181
42. Zhang DQ, Chen SC, Zhou Z-H (2005) A new face recognition method based on SVD perturbation for single example image per person. Appl Math Comput 163(2):895–907

43. Moon H (2004) Biometrics person authentication using projection-based face recognition system in verification scenario. In Proceedings of international conference on bioinformatics and its applications, Hong Kong, China, pp 207–213
44. Phillips PJ, Moon H, Rauss PJ, Rizvi SA (2000) The FERET evaluation methodology for face recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22:1090–1104
45. Choudhry T, Clarkson B, Jebara T, Pentland A (1999) Multimodal person recognition using unconstrained audio and video. In Proceedings of international conference on audio and video-based person authentication, pp 176–181
46. Chang C-Y, Chang C-W, Hsieh C-Y (2011) Applications of block linear discriminant analysis for face recognition. J Inf Hiding Multimedia Sig Process 2(3):259–269
47. Hastie T, Stuetzle W (1989) Principal curves. J Am Stat Assoc 84(406):502–516
48. Chang KY, Ghosh J (2001) A unified model for probabilistic principal surfaces. IEEE Trans Pattern Anal Mach Intell 23(1):22–41
49. Graepel T, Obermayer K (1999) A stochastic self-organizing map for proximity data. Neural Comput 11(1):139–155
50. Zhu Z, He H, Starzyk JA, Tseng C (2007) Self-organizing learning array and its application to economic and financial problems. Inf Sci 177(5):1180–1192
51. Yin H (2002) Data visualization and manifold mapping using the ViSOM. Neural Networks 15(8):1005–1016
52. Tenenbaum JB, Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
53. Roweis ST, Saul LK (2000) Nonlinear dimensionality deduction by locally linear embedding. Science 290(5500):2323–2326
54. He X, Niyogi P (2003) Locality preserving projections. In: Proceedings of conference on advances in neural information processing systems, pp 585–591
55. Manglem Singh KH (2011) Fuzzy rule based median filter for gray-scale images. J Inf Hiding Multimedia Sig Process 2(2):108–122
56. Zheng Z, Yang F, Tan W, Jia J, Yang J (2007) Gabor feature-based face recognition using supervised locality preserving projection. Sig Process 87(10):2473–2483
57. Zhu L, Zhu S (2007) Face recognition based on orthogonal discriminant locality preserving projections. Neurocomputing 70(7–9):1543–1546
58. Cai D, He X, Han J, Zhang HJ (2006) Orthogonal Laplacianfaces for face recognition. IEEE Trans Image Process 15(11):3608–3614
59. Yu X, Wang X (2008) Uncorrelated discriminant locality preserving projections. IEEE Sig Process Lett 15:361–364
60. Feng G, Hu D, Zhang D, Zhou Z (2006) An alternative formulation of kernel LPP with application to image recognition. Neurocomputing 69(13–15):1733–1738
61. Krinidis Stelios, Pitas Ioannis (2010) Statistical analysis of human facial expressions. J Inf Hiding Multimedia Sig Process 1(3):241–260
62. Kaganami HG, Ali SK, Zou B (2011) Optimal approach for texture analysis and classification based on wavelet transform and neural network. J Inf Hiding Multimedia Sig Process 2(1):33–40
63. Hu W-C, Yang C-Y, Huang D-Y, Huang Chun-Hsiang (2011) Feature-based face detection against skin-color like backgrounds with varying illumination. J Inf Hiding Multimedia Sig Process 2(2):123–132
64. Huang D-Y, Lin C-J, Hu W-C (2011) Learning-based face detection by adaptive switching of skin color models and AdaBoost under varying illumination. J Inf Hiding Multimedia Sig Process 2(3):204–216
65. Puranik P, Bajaj P, Abraham A, Palsodkar P, Deshmukh A (2011) Human perception-based color image segmentation using comprehensive learning particle swarm optimization. J Inf Hiding Multimedia Sig Process 2(3):227–235
66. Hu D, Feng G, Zhou Z (2007) Two-dimensional locality preserving projections (2DLPP) with its application to palmprint recognition. Pattern Recogn 40(1):339–342

67. Zhi R, Ruan Q (2008) Facial expression recognition based on two-dimensional discriminant locality preserving projections. Neurocomputing 71(7–9):1730–1734
68. Juwei Lu, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Networks 14(1):117–226
69. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
70. Liang Z, Shi P (2005) Uncorrelated discriminant vectors using a kernel method. Pattern Recogn 38:307–310
71. Liang Z, Shi P (2004) An efficient and effective method to solve kernel fisher discriminant analysis. Neurocomputing 61:485–493
72. Yang J, Frangi AF, Yang J-Y, Zhang D, Jin Z (2005) KPCA plus LDA: a complete kernel fisher discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27(2):230–244
73. Yang MH (2002) Kernel eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods. In: Proceedings of fifth IEEE international conference on automatic face and gesture recognition, pp 215–220
74. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Networks 14(1):117–126
75. Zheng W, Zou C, Zhao L (2005) Weighted maximum margin discriminant analysis with kernels. Neurocomputing 67:357–362
76. Huang J, Yuen PC, Chen W-S, Lai JH (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. In: Proceedings of the sixth IEEE international conference on automatic face and gesture recognition
77. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst Man Cybern Part B Cybern 35(3):556–562
78. Chen W-S, Yuen PC, Huang J, Dai D-Q (2005) Kernel machine-based one-parameter regularized fisher discriminant method for face recognition. IEEE Trans Syst Man Cybern Part B Cybern 35(4):658–669
79. Liang Y, Li C, Gong W, Pan Y (2007) Uncorrelated linear discriminant analysis based on weighted pairwise fisher criterion. Pattern Recogn 40:3606–3615
80. Zheng Y-J, Yang J, Yang J-Y, Wu X-J (2006) A reformative kernel fisher discriminant algorithm and its application to face recognition. Neurocomputing 69(13–15):1806–1810
81. Tao D, Tang X, Li X, Rui Y (2006) Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. IEEE Trans Multimedia 8(4):716–727
82. Xu Y, Zhang D, Jin Z, Li M, Yang J-Y (2006) A fast kernel-based nonlinear discriminant analysis for multi-class problems. Pattern Recogn 39(6):1026–1033
83. Saadi K, Talbot NLC, Cawley GC (2007) Optimally regularised kernel fisher discriminant classification. Neural Networks 20(7):832–841
84. Yeung D-Y, Chang H, Dai G (2007) Learning the kernel matrix by maximizing a KFD-based class separability criterion. Pattern Recogn 40(7):2021–2028
85. Shen LL, Bai L, Fairhurst M (2007) Gabor wavelets and general discriminant analysis for face identification and verification. Image Vis Comput 25(5):553–563
86. Ma B, Qu H-Y, Wong H-S (2007) Kernel clustering-based discriminant analysis. Pattern Recogn 40(1):324–327
87. Wu X-H, Zhou J–J (2006) Fuzzy discriminant analysis with kernel methods. Pattern Recogn 39(11):2236–2239
88. Liu Q, Lu H, Ma S (2004) Improving kernel Fisher discriminant analysis for face recognition. IEEE Trans Pattern Anal Mach Intell 14(1):42–49
89. Lanckriet G, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI (2004) Learning the kernel matrix with semidefinite programming. J Mach Learn Res 5:27–72
90. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Network 12(6):783–789

91. Tan X-Y, Chen S-C, Zhou Z-H et al (2006) Face recognition from a single image per person: a survey. Pattern Recogn 39(9):1725–1745
92. Malathi G, Shanthi V (2011) Statistical measurement of ultrasound placenta images complicated by gestational diabetes mellitus using segmentation approach. J Inf Hiding Multimedia Sig Process 2(4):332–343
93. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
94. Wu J, Zhou Z-H (2002) Face recognition with one training image per person. Pattern Recogn Lett 23(14):1711–1719
95. Chen SC, Zhang DQ, Zhou Z-H (2004) Enhanced (PC)2A for face recognition with one training image per person. Pattern Recogn Lett 25(10):1173–1181
96. Zhang DQ, Chen SC, Zhou Z-H (2005) A new face recognition method based on SVD perturbation for single example image per person. Appl Math Comput 163(2):895–907
97. Cox IJ, Ghosn J, Yianilos PN (1996) Feature based face recognition using mixture-distance. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 209–216
98. Lu X, Wang Y, Jain AK (2003) Combining classifiers for face recognition. In: Proceedings of IEEE international conference on multimedia and expo (ICME 2003), Baltimore, MD, pp 13–16
99. McCullagh D (2001) Call it super bowl face scan 1. Wired Magazine
100. Kim K (2005) Intelligent immigration control system by using passport recognition and face verification. In Proceedings of international symposium on neural networks, Chongqing, China, pp 147–156
101. Liu JNK, Wang M, Feng B (2005) iBotGuard: an internet-based intelligent robot security system using invariant face recognition against intruder. IEEE Trans Syst Man Cybern Part C Appl Rev 35:97–105
102. Acosta E, Torres L, Albiol A, Delp EJ (2002) An automatic face detection and recognition system for video indexing applications. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, vol 4. Orlando, Florida, pp 3644–3647
103. Lee J-H, Kim W-Y (2004) Video summarization and retrieval system using face recognition and MPEG-7 descriptors. Image and video retrieval, vol 3115. Lecture notes in computer science, Springer Berlin, Heidelberg, pp 179–188
104. Tredoux CG, Rosenthal Y, Costa LD, Nunez D (1999) Face reconstruction using a configural, eigenface-based composite system. In Proceedings of 3rd Biennial meeting of the society for applied research in memory and cognition (SARMAC), Boulder, Colorado, USA
105. Wijaya SL, Savvides M, Kumar BVKV (2005) Illumination-tolerant face verification of low-bitrate JPEG2000 wavelet images with advanced correlation filters for handheld devices. Appl Opt 44:655–665
106. Beymer D, Poggio T (1995) Face recognition from one example view. In: Proceedings of the international conference on computer vision, pp 500–507
107. González-Jiménez D, Alba-Castro JL (2007) Toward pose-invariant 2-D face recognition through point distribution models and facial symmetry. IEEE Trans Inf Forensic Secur 2(3–1):413–429
108. Cootes TF, Wheeler GV, Walker KN, Taylor CJ (2002) View-based active appearance models. Image Vis Comput 20:657–664
109. Kahraman F, Kurt B, Gokmen M (2007) Robust face alignment for illumination and pose invariant face recognition. In: Proceedings of the IEEE conference on CVPR, pp 1–7
110. Kakadiaris IA, Passalis G, Toderici G, Murtuza MN, Lu Y, Karampatziakis N, Theoharis T (2007) Three-dimensional face recognition in the presence of facial expressions: an annotated deformable model approach. IEEE Trans Pattern Anal Mach Intell 29(4):640–649
111. Gross R, Matthews I, Baker S (2004) Appearance-based face recognition and light-fields. IEEE Trans Pattern Anal Mach Intell 26(4):449–465

112. Liu C (2004) Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Trans Pattern Anal Mach Intell 26(5):572–581
113. Xie X, Lam KM (2006) Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image. IEEE Trans Image Process 15(9):2481–2492
114. Huang J, Yuen PC, Chen WS, Lai JH (2007) Choosing parameters of kernel subspace LDA for recognition of face images under pose and illumination variations. IEEE Trans Syst Man Cybern B Cybern 37(4):847–862
115. Yang J, Frangi AF, Yang J, Zhang D, Jin Z (2005) KPCA plus LDA: a complete kernel fisher discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27(2):230–244
116. Kim TK, Kittler J (2006) Design and fusion of pose-invariant face-identification experts. IEEE Trans Circuits Syst Video Technol 16(9):1096–1106
117. Levine MD, Yu Y (2006) Face recognition subject to variations in facial expression, illumination and pose using correlation filters. Comput Vis Image Underst 104(1):1–15
118. Chai X, Shan S, Chen X, Gao W (2007) Locally linear regression for pose-invariant face recognition. IEEE Trans Image Process 16(7):1716–1725
119. Prince SJD, Warrell J, Elder JH, Felisberti FM (2008) Tied factor analysis for face recognition across large pose differences. IEEE Trans Pattern Anal Mach Intell 30(6):970–984
120. Kirby M, Sirovich L (1990) Application of the Karhunen–Loève procedure for the characterization of human face. IEEE Trans Pattern Anal Mach Intell 12(1):103–108
121. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
122. Turk MA, Pentland AP (1991) Face recognition using eigenfaces. In: Proceedings of the IEEE conference on CVPR, pp 586–591
123. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. IEEE Trans Neural Network 8(1):98–113
124. Gao Y, Leung MKH (2002) Face recognition using line edge map. IEEE Trans Pattern Anal Mach Intell 24(6):764–779
125. Gao Y, Qi Y (2005) Robust visual similarity retrieval in single model face databases. Pattern Recogn 38:1009–1020
126. Gao Y, Leung MKH, Wang W, Hui SC (2001) Fast face identification under varying pose from a single 2-D model view. IEE Proc Vis Image Sig Process 148(4):248–253
127. Liu X, Chen T (2005) Pose-robust face recognition using geometry assisted probabilistic modeling. In: Proceedings of the IEEE conference on CVPR, vol 1, pp 502–509
128. Zhang X, Gao Y, Leung MKH (2006) Automatic texture synthesis for face recognition from single views. In: Proceedings of the ICPR, vol 3, pp 1151–1154
129. Lee MW, Ranganath S (2003) Pose-invariant face recognition using a 3D deformable model. Pattern Recogn 36(8):1835–1846
130. Jiang D, Hu Y, Yan S, Zhang L, Zhang H, Gao W (2005) Efficient 3D reconstruction for face recognition. Pattern Recogn 38(6):787–798
131. Zhang X, Gao Y, Leung MKH (2008) Recognizing rotated faces from frontal and side views: an approach towards effective use of mug shot databases. IEEE Trans Inf Forensic Secur 3(4):684–697
132. Blanz V, Vetter T (1999) A morphable model for the synthesis of 3D faces. In: Proceedings of SIGGRAPH, pp 187–194
133. Blanz V, Vetter T (2003) Face recognition based on fitting a 3D morphable model. IEEE Trans Pattern Anal Mach Intell 25(9):1063–1074
134. Georghiades AS, Belhumeur PN, Kriegman DJ (2000) From few to many: generative models for recognition under variable pose and illumination. In: Proceedings of the international conference on auto face gesture recognition, pp 277–284
135. Georghiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660

136. Castillo CD, Jacobs DW (2007) Using stereo matching for 2-D face recognition across pose. In: Proceedings of the IEEE conference on CVPR, pp 1–8
137. Beymer DJ (1994) Face recognition under varying pose. In: Proceedings of the IEEE conference on CVPR, pp 756–761
138. Singh R, Vatsa M, Ross A, Noore A (2007) A mosaicing scheme for pose-invariant face recognition. IEEE Trans Syst Man Cybern B Cybern 37(5):1212–1225
139. Brunelli R, Poggio T (1993) Face recognition: features versus templates. IEEE Trans Pattern Anal Mach Intell 15(10):1042–1052
140. Pentland A, Moghaddam B, Starner T (1994) View-based and modular eigenspaces for face recognition. In: Proceedings of the IEEE conference on CVPR, pp 84–91
141. Wiskott L, Fellous JM, Kruger N, vonder Malsburg C (1997) Face recognition by elastic bunch graph matching. IEEE Trans Pattern Anal Mach Intell 19(7):775–779
142. Ahonen T, Hadid A, Pietikainen M (2006) Face description with local binary patterns: application to face recognition. IEEE Trans Pattern Anal Mach Intell 28(12):2037–2041
143. Manjunath BS, Chellappa R, Malsburg CVD (1992) A feature based approach to face recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition 1(1992), pp 373–378
144. Lades M, Vorbruggen J, Buhmann J, Lange J, von der Malsburg C, Wurtz R (1993) Distortion invariant object recognition in the dynamic link architecture. IEEE Trans Comput 42(3):300–311
145. Wiskott L, Fellous JM, Kruger N, von Malsburg C (1997) Face recognition by elastic bunch graph matching. IEEE Trans Pattern Anal Mach Intell 19(7):775–779
146. Kepenekci B, Tek FB, Akar GB (2002) Occluded face recognition based on Gabor wavelets. In: Proceedings of ICIP 2002, MP-P3.10, Sep 2002, Rochester, NY
147. Gao Y, Qi Y (2005) Robust visual similarity retrieval in single model face databases. Pattern Recogn 38(7):1009–1020
148. Chen SC, Liu J, Zhou Z-H (2004) Making FLDA applicable to face recognition with one sample per person. Pattern Recogn 37(7):1553–1555
149. Huang J, Yuen PC, Chen WS, Lai JH (2003) Component-based LDA method for face recognition with one training sample. In: Proceedings of AMFG (2003), pp 120–126
150. Lawrence S, Giles CL, Tsoi A, Back A (1997) Face recognition: a convolutional neural-network approach. IEEE Trans Neural Networks 8(1):98–113
151. Tan X, Chen SC, Zhou Z-H, Zhang F (2005) Recognizing partially occluded, expression variant faces from single training image per person with SOM and soft kNN ensemble. IEEE Trans Neural Networks 16(4):875–886
152. Le H-S, Li H (2004) Recognizing frontal face images using hidden Markov models with one training image per person. In: Proceedings of the 17th international conference on pattern recognition (ICPR04), vol 1, pp 318–321
153. Martinez AM (2002) Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE Trans Pattern Anal Mach Intell 25(6):748–763
154. Komleh HE, Chandran V, Sridharan S (2001) Robustness to expression variations in fractal-based face recognition. In: Proceedings of ISSPA-01, vol 1, Kuala Lumpur, Malaysia, 13–16 Aug 2001, pp 359–362
155. Lam K-M, Yan H (1998) An analytic-to-holistic approach for face recognition based on a single frontal view. IEEE Trans Pattern Anal Mach Intell 20(7):673–686
156. Ahonen T, Hadid A, Pietikäinen M (2004) Face recognition with local binary patterns. In: Proceedings of computer vision, ECCV 2004, lecture notes in computer science 3021, Springer, pp 469–481
157. Jung HC, Hwang BW, Lee SW (2004) Authenticating corrupted face image based on noise model. In: Proceedings of the sixth IEEE international conference on automatic face and gesture recognition, p 272
158. Wu J, Zhou Z-H (2002) Face Recognition with one training image per person. Pattern Recogn Lett 23(14):1711–1719

159. Wang J, Plataniotis KN, Venetsanopoulos AN (2005) Selecting discriminant eigenfaces for face recognition. Pattern Recogn Lett 26(10):1470–1482
160. Yang J, Zhang D, Frangi AF, Yang J (2004) Two-dimensional PCA: a new approach to appearance based face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26(1):131–137
161. De la Torre F, Gross R, Baker S, Kumar V (2005) Representational oriented component analysis (ROCA) for face recognition with one sample image per training class. In: Proceedings of IEEE conference on computer vision and pattern recognition 2(June 2005), pp 266–273
162. Chen SC, Zhang DQ, Zhou Z-H (2004) Enhanced (PC)2A for face recognition with one training image per person. Pattern Recogn Lett 25(10):1173–1181
163. Vetter T (1998) Synthesis of novel views from a single face image. Int J Comput Vis 28(2):102–116

Chapter 3
Kernel Learning Foundation

3.1 Introduction

Nonlinear information processing algorithms can be designed by means of linear techniques in implicit feature spaces induced by kernel functions. Kernel methods are algorithms that, by replacing the inner product with an appropriate positive definite function, implicitly perform a nonlinear mapping of the input data to a high-dimensional feature space. This idea can be traced back to the potential function method [1, 2], and it has been successfully applied to the support vector machine (SVM), a learning method with controllable capacity which obtains outstanding generalization in high (even infinite)-dimensional feature spaces [3–7]. The kernel method can be used if the interactions between elements of the domain occur only through inner products. This suggests the possibility of building nonlinear, kernel-based counterparts of standard pattern analysis algorithms. Recently, a nonlinear feature extraction method has been presented [8] based on a kernel version of principal component analysis (PCA) and [9] proposed a nonlinear kernel version of Fisher discriminant analysis. Kernel-based versions of other pattern analysis algorithms have been also proposed in [10, 11], among others; the field of the so-called kernel machines is now extremely active [12]. While powerful kernel methods have been proposed for supervised classification and regression problems, the development of effective kernel method for clustering, aside for a few tentative solutions, is still an open problem. One-class SVM characterizes the support of a high-dimensional distribution. Intuitively, one-class SVM computes the smallest sphere in feature space enclosing the image of the input data. Saha et al. [13] also integrated the SVR with other statistical methods to improve the ability of system prognostics. Zio et al. [14] proposed a similarity-based prognostics to estimate remaining useful life (RUL) of the system, and this method applied fuzzy similarity analysis on the evolution data of the reference trajectory patterns. Similar to the SVM, the relevance vector machine (RVM) [15] is a type of improved machine learning algorithm based on Bayesian framework. SVR algorithm is a machine learning method based on statistical learning theory and widely applied in time series prediction. It can also be used for status

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_3,

monitoring and forecasting. However, SVR cannot meet the demands of online and real-time application because of the time-consuming computation. In the same way, most of the traditional offline forecasting data-driven methods are facing the same challenge. Hence, many training algorithms, such as incremental algorithm and decremental algorithm [16, 17], have been proposed to update these methods to online style and further decrease the computing complexity. Incremental training algorithm is the most popular online learning strategy for the SVR [16–24]. In paper [3], the author proposed the accurate incremental learning algorithm and applied the new algorithm in the classification of SVM. Furthermore, Ma et al. [21] introduced the accurate online SVR algorithms based on the incremental algorithm to solve the approximation and regression problems. Lots of researchers proposed many improved online SVR algorithm for different applications [25–29]. Although all the online algorithms above can achieve the model dynamically learning with the sample updating, the computing complexity was very high. Moreover, in some online prediction methods such as online SVR, conflicts and trade-offs between prediction efficiency and accuracy still exist. A new approach included five improved online SVR algorithms previously proposed in our research that are applied to realize adaptive online status monitoring and fault prognostics. Experimental results with the Tennessee Eastman process fault data as well as mobile traffic data show that the algorithms can be effectively applied to the online status prediction with excellent performance in both efficiency and precision.

3.2 Linear Discrimination and Support Vector Machine

Suppose that a set of N entities in the feature space $X = \{x_0, x_1, x_2, \ldots, x_p\}$ is partitioned into two classes and a function $u = f(x_0, x_1, x_2, \ldots, x_p)$ that would discriminate the two classes. To find an appropriate w, even in the case when classes are linearly separable, various criteria can be utilized. A most straightforward classifier is defined by the least-squares criterion. This produces

$$w = \left(X^T X\right)^{-1} X^T u \tag{3.1}$$

Note that formula leads to the different criterion of minimizing the ratio of the within-class error to out-of-class error. The least-squares linear decision rule based on Bayes function is construct the covariance Gaussian matrix, the rule is described with the following formula:

$$fi(x) = \exp\left[-(x - \mu_i)^T \sum{}^{-1} (x - \mu_i)/2\right] / \left[(2p)^p \left|\sum\right|\right]^{1/2} \tag{3.2}$$

SVM was first proposed in 1992. As supervised learning methods, SVMs are used for classification and regression. SVMs use machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data [30]. SVM becomes famous when using pixel maps as input; it gives accuracy

comparable with sophisticated neural networks with elaborated features in a handwriting recognition task [31]. It is also being used for many applications, such as handwriting analysis [32] and structural risk minimization (SRM) [33]. And it is used with traditional empirical risk minimization (ERM) principle. SVMs were developed to solve regression problems [34].

Given a training set of N data points $\{y_k, x_k\}_{k=1}^{N}$, where $x_k \in R^n$ is the kth input pattern and $y_k \in R$ is the kth output pattern, the classifier can be constructed using the support vector method in the form

$$y(x) = \text{sign}\left[\sum_{k=1}^{N} \alpha_k y_k K(x, x_k) + b\right] \tag{3.3}$$

where α_k are called support values and b is a constant. The $K(\cdot,\cdot)$ is the kernel, which can be either $K(x, x_k) = x_k^T x$ (linear SVM); $K(x, x_k) = \left(x_k^T x + 1\right)^d$ (polynomial SVM of degree d); $K(x, x_k) = \tanh[\kappa x_k^T x + \theta]$ (multilayer perceptron SVM); or $K(x, x_k) = \exp\left\{-\|x - x_k\|_2^2/\sigma^2\right\}$ (RBF SVM).

For instance, the problem of classifying two classes is defined as

$$\begin{cases} w^T\phi(x_k) + b \geq +1 & \text{if } y_k = +1 \\ w^T\phi(x_k) + b \leq -1 & \text{if } y_k = -1 \end{cases} \tag{3.4}$$

This can also be written as

$$y_k\left[w^T\phi(x_k) + b\right] \geq 1, \quad k = 1, \ldots, N \tag{3.5}$$

where $\phi(\cdot)$ is a nonlinear function mapping of the input space to a higher-dimensional space. LS-SVM classifiers

$$\min_{w,b,e} J_{\text{LS}}(w, b, e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^{N} e_k^2 \tag{3.6}$$

subject to the equality constraints

$$y_k\left[w^T\phi(x_k) + b\right] = 1 - e_k, \quad k = 1, \ldots, N \tag{3.7}$$

The Lagrangian is defined as

$$L(w, b, e; \alpha) = J_{\text{LS}} - \sum_{k=1}^{N} \alpha_k \left\{y_k\left[w^T\phi(x_k) + b\right] - 1 + e_k\right\} \tag{3.8}$$

with Lagrange multipliers $\alpha_k \in R$ (called support values). The conditions for optimality are given by

$$\begin{cases} \frac{\partial L}{\partial w} = 0 & \rightarrow & w = \sum_{k=1}^{N} \alpha_k y_k \phi(x_k) \\ \frac{\partial L}{\partial b} = 0 & \rightarrow & \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial L}{\partial e_k} = 0 & \rightarrow & \alpha_k = \gamma\, e_k \\ \frac{\partial L}{\partial \alpha_k} = 0 & \rightarrow & y_k[w^T \phi(x_k) + b] - 1 + e_k = 0 \end{cases} \tag{3.9}$$

For $k = 1, \ldots, N$. After elimination of w and e, one obtains the solution

$$\begin{bmatrix} 0 & Y^T \\ Y & ZZ^T + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix} \tag{3.10}$$

with $Z = \left[\phi(x_1)^T y_1; \ldots; \phi(x_N)^T y_N\right]$, $Y = [y_1; \ldots; y_N]$, $1_v = [1; \ldots; 1]$, $e = [e_1; \ldots; e_N]$ and $\alpha = [\alpha_1; \ldots; \alpha_N]$. Mercer's condition is applied to the matrix $\Omega = ZZ^T$ with

$$\Omega_{kl} = y_k y_l \phi(x_k)^T \phi(x_l) = y_k y_l K(x_k, x_l) \tag{3.11}$$

3.3 Kernel Learning: Concepts

Kernel: If the data are linear, you can use the division to separate hyperplane data. Often, however, the situation is that the data are from a linear datasets are inseparable. The nonlinear mapping of input data is allowed to computed with kernel function in a high-dimensional data space. Then, the new map is linearly separable. Here is a very simple illustration of this point. This mapping is defined by the kernel:

$$K(x, y) = \langle \Phi(x) \cdot \Phi(y) \rangle \tag{3.12}$$

The kernel function value of two samples can be computed with the dot product.

$$\langle x_1 \cdot x_2 \rangle \leftarrow K(x_1, x_2) = \langle \Phi(x_1) \cdot \Phi(x_2) \rangle \tag{3.13}$$

Note the legend is not described as they are sample plotting to make understand the concepts involved.

Polynomial: A polynomial mapping is a popular method for nonlinear modeling. The second kernel is usually preferable as it avoids problem with the hessian becoming zero.

$$k(x, y) = (x \cdot y)^d \quad (d \in N) \tag{3.14}$$

Gaussian Radial Basis Function: Radial basis functions is most commonly used with a Gaussian form. A radial basis function (RBF) produces a piecewise linear solution which can be attractive when discontinuities are acceptable.

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \quad (\sigma > 0) \tag{3.15}$$

Multi-Layer Perceptron: The long-established MLP, with a single hidden layer, also has a valid kernel representation.

$$k(x, z) = \tanh(\alpha < x, z > + \beta), (\alpha > 0, \beta < 0) \tag{3.16}$$

There are many more, including Fourier, splines, B-splines, additive kernels, and tensor products.

3.4 Kernel Learning: Methods

3.4.1 Kernel-Based HMMs

Hidden Markov models (HMMs) are designed for recognition tasks in which the observation is often a single measurement. However, when HMMs are applied to dynamic sequence synthesis, the observations are multiple sequences consisting of both input and output. Therefore, the conventional HMMs have to be extended for the purpose of dynamic input/output mapping. Another drawback of simply applying a traditional HMM to synthesis is that one has to specify the parametric form of the observation model, such as a Gaussian. In many cases, such parametric models only capture the global characteristics of the observation distribution. Since synthesis can be regarded as an inverse problem of density estimation, i.e., finding a sample that satisfies a given probabilistic distribution, a parametric model is often too smooth and/or uniform to capture the fine details.

A commonly used method to model the mapping between two sequences is regression. For example, a neural-network-based approach is used in [33] to learn the mapping between lip motion and speech. But in many cases, the mapping relations between input and output is many to many, which defeats the classical regression methods, even with a local window context. As a powerful approach to model time series data in the state space, HMMs have been adopted in many synthesis applications [35, 36]. Previous approaches assume that either the input or the output can be modeled by an HMM. For example, the video rewrite [36] technique recognizes different phonemes from the input audio signal. Animation is generated by reordering the captured video frames which share similar phonemes as in the training video. On the other hand, the shadow puppetry technique trains an HMM model to synthesize 3D motions from observed 2D images. A remapping process is employed to give each state a dual mapping into both 3D motion and 2D silhouette.

The underlying assumption made in previous approaches is that the input sequence shares the dynamic behavior exhibited in the HMM trained from the output or vice versa [36]. As a result, the output has no ability to directly adapt to the input. Although some more complex HMMs have been proposed for multiple observations or multiple state sequences, none of them has been specifically designed for synthesis. For example, factorial HMM uses a more complex state structure to improve the representational capacity of the HMM [37]. More efficient observation models are designed to represent a set of dynamic sequences with a single HMM [38]. For dynamic sequence synthesis, we expect the result to keep the patterns in the training data, but not to replicate exactly the same data. In traditional HMMs, state observation is generalized by some parametric models (e.g., Gaussian mixtures), which are too smooth and/or uniform to capture fine details. This problem has been addressed in motion texture [39] by introducing a nonparametric probabilistic model. However, since some manually labeled phases are involved in this approach, it would be ineffective when the dynamics of the synthesized sequence are complex. Recently, switched linear dynamic system [40] is used for learning and synthesizing human motions. But in dynamic sequence synthesis, a number of sublinear systems are required to approximate the highly nonlinear relations between the input and output, which makes the learning and synthesis intractable.

Although sample sets are sufficient to approximate continuous-valued distributions, it is difficult to evaluate in the EM algorithm. Therefore, we need to replace the Dirac function with a continuous function so that the resulting density is continuous.

3.4.2 Kernel-Independent Component Analysis

Kernel-independent component analysis (ICA) in the kernel-inducing feature space is to develop a two-phase kernel ICA algorithm: whitened kernel principal component analysis (KPCA) plus ICA. KPCA spheres data and makes the data structure become as linearly separable as possible by virtue of an implicit nonlinear mapping determined by kernel. ICA seeks the projection directions in the KPCA whitened space, making the distribution of the projected data as non-Gaussian as possible.

Over the last few years, ICA has aroused wide research interests and become a popular tool for blind source separation and feature extraction. From the feature extraction point of view, ICA has a close relationship with projection pursuit since both of them aim to find the directions such that the projections of the data into those directions have maximally “non-Gaussian” distributions. These projections are interesting and considered more useful for classification [41]. Bartlett [42] and Liu [43] applied ICA to face representation and recognition and found that it outperforms PCA when cosine distance was used as the similarity measure. ICA, however, fails to separate the nonlinearly mixed source due to its intrinsic

linearity. Likewise, for feature extraction, ICA-based linear projection is incompetent to represent the data with nonlinear structure. To address this problem, our idea is to nonlinearly map the data into a feature space, in which the data have a linear structure (as linearly separable as possible). Then, we perform ICA in feature space and make the distribution of data as non-Gaussian as possible. We will use "kernel tricks" to solve the computation of independent projection directions in high-dimensional feature space and ultimately convert the problem of performing ICA in feature space into a problem of implementing ICA in the KPCA transformed space. It should be mentioned that kernel ICA formulation in this book is different from that in [5]. In reference [44], the kernel trick is used for computation and optimization of a canonical correlation-based contrast function, while in this book, "kernel" is introduced to realize an implicit nonlinear mapping, which makes the data linearly structured in feature space.

Given a random vector $\boldsymbol{x}$, which is possibly nonlinearly mixed, we map it into its image in the feature space H by the following nonlinear mapping: As a result, a pattern in the original observation space (input space) Rn is mapped into a potentially much higher-dimensional feature vector in the *feature space H*. Assume that after the nonlinear mapping, the data have a linearly separable structure in feature space H.

Let us recall the implementation of ICA in observation space. Before applying an ICA algorithm on the data, it is usually very useful to do some preprocessing work (e.g., sphering or whitening data). The preprocessing can make the problem of ICA estimation simpler and better conditioned. Similarly, we can perform PCA in feature space for data whitening. Note that performing PCA in feature space can be equivalently implemented in input space (observation space) by virtue of kernels, i.e., performing KPCA based on the observation data. Note that the new unmixing matrix W should be orthogonal. In summary, the ICA transformation in feature space can be decomposed into two: the whitened KPCA transformation in input space and the common ICA transformation in the KPCA whitened space.

3.5 Kernel-Based Online SVR

Given a status monitoring data as time series training set, $T = \{(x_1, y_1), \cdots, (x_l, y_l)\} \in (X \times Y)^l$, where $x_i \in X = R^n$, $y_i \in Y = R$, $i = 1, \cdots l$, the main task of SVR is to construct a linear regression function,

$$f(x) = W^T \phi(x) + b \tag{3.17}$$

in feature space F. W is a vector in F, and $\phi(x)$ maps the input x to a vector in F.

The W and b in Eq. (3.1) are obtained by solving an optimization problem:

$$\begin{aligned} &\min_{w,b} P = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*). \\ &s.t.\ ((w \cdot x_i) + b) - y_i \le \varepsilon + \xi_i, \quad i = 1, 2, \ldots, l, \\ &\quad y_i - ((w \cdot x_i) + b) \le \varepsilon + \xi_i, \quad i = 1, 2, \ldots, l, \\ &\quad \xi_i^* \ge 0, \quad i = 1, 2, \ldots, l, \end{aligned} \tag{3.18}$$

Here, the slack variables ξ_i and ξ_i^*, penalty parameter C, ε-insensitive loss. Convert the responding Lagrangian as

$$\begin{aligned} &\min_{\alpha,\alpha^i} \frac{1}{2}\sum_{i}^{l}\sum_{j}^{l} Q_{ij}(\alpha_i - \alpha_i^*)\left(\alpha_j - \alpha_j^*\right) + \varepsilon\sum_{i}^{l}(\alpha_i + \alpha_i^*) - \sum_{i}^{l} y_i(\alpha_i - \alpha_i^*) \\ &s.t.\ \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0, \\ &\quad 0 \le \alpha_i, \alpha_i^* \le \frac{C}{l}, \quad i = 1, 2, \ldots, l \end{aligned} \tag{3.19}$$

Define kernel function: $Q_{ij} = \phi(x_i)^T\phi(x_j) = K(x_i, x_j)$; then, the regression function can be described as

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(x_i, x) + b \tag{3.20}$$

Due to Karush–Kuhn–Tucker (KKT) conditions, we can get

$$\begin{cases} h(x_i) \ge \varepsilon, & \theta_i = -C \\ h(x_i) = \varepsilon, & -C < \theta_i < 0 \\ -\varepsilon \le h(x_i) \le \varepsilon, & \theta_i = 0 \\ h(x_i) = \varepsilon, & 0 < \theta_i < C \\ h(x_i) \le -\varepsilon, & \theta_i = C \end{cases} \tag{3.21}$$

here $\theta_{\mathrm{i}} = \alpha_i - \alpha_i^*$, $h(x) \equiv f(x_i) - y_i = \sum_{j=1}^{l} Q_{\mathrm{ij}}\theta_j - y_i + b$.

Depending on the sign of $f(x_i) - y_i$, we can get

(a) The E set: $E = \{i|\ |\theta_i| = C\}$
(b) The S set: $S = \{i|\ 0 < |\theta_i| < C\}$
(c) The R set: $R = \{i|\ \theta_i = 0\}$.

Batch SVR retrains the model when the data update. With the retraining of SVR each time, it brings the problems of low speed and inefficiency, while the new sample adds. The online SVR trains with incremental algorithm and decremental algorithm when dataset updates. The structure of online SVR adjusts dynamically to meet the KKT conditions [31, 38].

Let x_c be a new training sample, corresponding define θ_c, then compute $\theta_i (i = 1, 2, \ldots, n)$, $\Delta\theta_i$ and $\Delta\theta_c$ to meet the KKT conditions.

$$\Delta h(x_i) = K(x_i, x_c)\Delta\theta_c + \sum_{i=1}^{n} K(x_i, x_j)\Delta\theta_j + \Delta b \tag{3.22}$$

$$\theta_c + \sum_{i=1}^{n} \theta_i = 0 \tag{3.23}$$

For S set,

$$\begin{aligned} &\sum_{j\in S} K(x_i, x_j)\Delta\theta_j + \Delta b = -K(x_i, x_c)\Delta\theta_c \\ &\sum_{j\in S} \Delta\theta_j = -\Delta\theta_c \\ &i \in S \end{aligned} \tag{3.24}$$

Equation 3.8 can be represented in matrix as

$$\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(x_{s_1}, x_{s_1}) & \cdots & K(x_{s_1}, x_{s_{l_s}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_{s_{l_s}}, x_{s_{l_1}}) & \cdots & K(x_{s_{l_s}}, x_{s_{l_s}}) \end{bmatrix} \begin{bmatrix} \Delta b \\ \Delta\theta_{s_1} \\ \vdots \\ \Delta\theta_{s_{l_s}} \end{bmatrix} = -\begin{bmatrix} 1 \\ K(x_{s_1}, x_c) \\ \vdots \\ K(x_{s_{l_s}}, x_{s_{l_s}}) \end{bmatrix} \Delta\theta_c \tag{3.25}$$

From Eq. 3.9, we can get

$$\begin{bmatrix} \Delta b \\ \Delta\theta_{s_1} \\ \vdots \\ \Delta\theta_{s_{l_s}} \end{bmatrix} = -\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K(x_{s_1}, x_{s_1}) & \cdots & K(x_{s_1}, x_{s_{l_s}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_{s_{l_s}}, x_{s_{l_1}}) & \cdots & K(x_{s_{l_s}}, x_{s_{l_s}}) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ K(x_{s_1}, x_c) \\ \vdots \\ K(x_{s_{l_s}}, x_{s_{l_s}}) \end{bmatrix} \Delta\theta_c = \beta\Delta\theta_c \tag{3.26}$$

then

$$\begin{bmatrix} \Delta h(x_{n_1}) \\ \Delta h(x_{n_2}) \\ \vdots \\ \Delta h(x_{n_{l_N}}) \end{bmatrix} = \left\{ \begin{bmatrix} K(x_{n_1}, x_c) \\ K(x_{n_2}, x_c) \\ \vdots \\ K(x_{n_{l_N}}, x_c) \end{bmatrix} + \begin{bmatrix} 1 & K(x_{n_1}, x_{s_1}) & \cdots & K(x_{n_1}, x_{s_{l_s}}) \\ 1 & K(x_{n_2}, x_{s_1}) & \cdots & K(x_{n_2}, x_{s_{l_s}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K(x_{n_{l_N}}, x_c) & \cdots & K(x_{n_{l_N}}, x_{s_{l_s}}) \end{bmatrix} \beta \right\} \tag{3.27}$$

The online SVR implements the update of the S set, the E set, and the R set. While deleting a training sample accordingly, the computation process is similar.

The detail of computation is discussed in book. Through the training process described above, the online SVR implements the update of the S set, the E set, and the R set without retraining all the dataset.

3.6 Optimized Kernel-Based Online SVR

Five improved and optimized online SVR algorithms are introduced according to a variety of prediction needs and datasets with various features. Accordingly, considering the influence of kernel function types and sample scales on the online SVR algorithm, we focus on improving the precision and efficiency of online time series prediction with kernel combination and sample reduction. The five types of improved online SVR algorithms can be classified into two categories:

1. Kernel-combination-based optimized online SVR: Method I and Method II.
2. Sample-reduction-based optimized online SVR: Method III, Method IV, and Method V.

Therefore, in order to achieve complex equipment or system online monitoring and status prediction, we can select one of those algorithms or combined different models under the different application conditions of fault prognostics. The five improved online SVR algorithms which are previously proposed to achieve trade-off between prediction precision and efficiency are described as follows.

3.6.1 Method I: Kernel-Combined Online SVR

1. **Analysis of Kernel Types of Online SVR**

The most important work for online SVR is the choice of kernel function. For complicated and nonstationary nonlinear status prediction, suitable kernel function should be adopted according to the features of the complicated data. There are two types of kernel functions: global kernel and local kernel, whose characteristics are different.

The learning ability of the local kernel is strong, but the generalization performance is weak, and at the same time, the learning ability of the global kernel is weak, but the generalization performance is strong. Keerthi [1, 7] proved that the RBF kernel function can replace the polynomial kernel function with the choice of appropriate parameters. For most offline applications, sufficient prior knowledge, plenty of analysis for data samples, and the selection of the appropriate type of kernel function and its parameters can be obtained in advance. Therefore, the RBF kernel is used for modeling and the result would not be too bad under this offline condition.

However, because of unpredictable online updated data series and online modeling data length restrictions, it can be of difficulty to select some kind of kernel function for online modeling and forecasting. The data with long distance influenced the value of the global kernel functions, while only the data in neighborhood decide the value of the local kernel almost. Linear kernel and polynomial kernel are global, while RBF kernel and Gaussian kernel are local for SVR.

For general fault modes, the failure is usually slowly varied process for the apparatuses and systems. Nevertheless, some unexpected failures belong to occasional or emergent conditions. Either the global kernel or local kernel could not be applicable to predict all fault status independently. The complexity of failures gives the conclusion that global kernel should be used for the fault prediction because the system fault was influenced by all running process data. In addition, the local kernel is employed for some casually happened fault due to accidental factor only related to the data in local domain. In other words, single global or local kernel could not satisfy multiple fault modes.

The KCO-SVR combines different characteristics of global and local kernel functions to fit the diversity of the complicated fault datasets. Therefore, this algorithm is more suitable and convenient for prognostics than the single kernel online SVR.

2. **Proposed KCO-SVR algorithm**

Both global and local kernel functions should be used in fault prediction through the analysis of the failure modes to make better prediction. So we proposed a new online SVR algorithm that combined two different types of kernels named kernel-combined online support vector regression (KCO-SVR). With the combination of two types of kernels, the combined online prediction could realize better result for the complicated time series. The trend could be fitted by the global kernel for those samples far from the historical data, and the correlation in the neighborhood could be fitted by the local kernel. The combination could supplement the drawback of the single-type kernel in the prediction for the complex time series.

In the KCO-SVR, the decision function represented that the global characteristic is first obtained with the global kernel, as

$$f_g(x) = \sum_{i=1}^{m} \left(\alpha_{gi} - \alpha_{gi}^*\right) K_g\left(x_{gi}, x_g\right) + b_g \tag{3.28}$$

Here, the support vector set $K_g\left(x_{gi}, x_g\right) \in S_g$ is obtained by training with the global kernel. And the local features are contained in the variance of the global modeling.

The variance of the time series is modeled by the local kernel online SVR, and we can get the corresponding decision function:

$$f_l(x) = \sum_{j=1}^{k} \left(\alpha_{lj} - \alpha_{lj}^*\right) K_l\left(x_{lj}, x_l\right) + b_l \tag{3.29}$$

Here, the support vector set $K_l(x_{lj}, x_l) \in S_l$ is obtained by training with the local kernel.

So the final decision/prediction function is as follows:

$$f(x) = \sum_{i=1}^{m} \left(\alpha_{gi} - \alpha_{gi}^*\right) K_g(x_{gi}, x_g) + b_g + \sum_{j=1}^{k} \left(\alpha_{lj} - \alpha_{lj}^*\right) K_l(x_{lj}, x_l) + b_l \tag{3.20}$$

The flow of the KCO-SVR for time series prediction is as follows.

Definition Online SVR_1(global kernel), online SVR_2(local kernel), C_1, C_2(penalty parameters), ε_1, ε_2(ε-insensitive loss), p_1 , p_2(kernel parameters), *train length* (the length of the online training dataset), *embedded dimension* (embedded dimension);

Output: prediction result *PredictL(i)*;
The detailed steps are as follows:

a. Data preprocessing;
b. Online SVR_1 initial training with global kernel such as linear and polynomial kernel functions;
c. Online SVR_2 initial training with local kernel such as RBF kernel functions, and the modeling dataset is the variance time series in the step (b);
d. For updating online time series (x_c, y_c), realize online training for the online SVR_1 and online SVR_2 with incremental algorithms;
e. Online SVR_1 output prediction result $\widetilde{G}(k+1)$;
f. Online SVR_2 output prediction result $\widetilde{L}(k+1)$;
g. Combine the prediction results $\widetilde{X}(k+1) = \widetilde{G}(k+1) + \widetilde{L}(k+1)$ or sum them with certain weights;
h. Update the online SVR_1 and online SVR_2 with decremental algorithm;
i. Dataset online update, repeat step (d) to step (h).

3.6.2 Method II: Local Online Support Vector Regression

The KCO-SVR algorithm combined the two types of kernels to increase the prediction accuracy compared with the single kernel online SVR in online data forecasting. However, the use of different kernel functions together would get low operation efficiency. So we proposed a new online SVR algorithm by combining the offline and online models with different types of kernels, which is named as variance prediction compensation local online support vector regression (VPCLO-SVR).

Therefore, in order to improve the efficiency of combined forecasting method, we consider to replacing the global kernel online SVR algorithm as the offline

algorithm. With the local offline SVR algorithm, the online combined models became simple and fast.

Here, we replace the online SVR in the Eq. (3.12) with offline batch SVR algorithm and the global decision function is obtained with offline SVR with batch training. As a result, the long-term trend could be represented by the decision function.

$$f_{\mathrm{Bg}}(x) = \sum_{i=1}^{m} \left(\alpha_{\mathrm{Bgi}} - \alpha^{*}_{\mathrm{Bgi}}\right) K_{\mathrm{Bg}}\left(x_{\mathrm{Bgi}}, x_{\mathrm{Bg}}\right) + b_{\mathrm{Bg}} \tag{3.31}$$

Here, the support vector set $K_{\mathrm{Bg}}\left(x_{\mathrm{Bgi}}, x_{\mathrm{Bg}}\right) \in S_{\mathrm{Bg}}$ is obtained by training the global offline kernel function.

Consequently, the VPCLO-SVR used the global kernel to fit and approximate trend properties of time series. Then, the predicting residual with the real data stream is calculated. Finally, the residual for the predicted value is predicted on online SVR to compensate predicted value with the local SVR. Compared to KCO-SVR algorithm, the VPCLO-SVR algorithm can efficiently improve the efficiency while keeping the prediction precision.

The flow of the VPCLO-SVR for time series prediction is as follows.

Definition Batch SVR (global kernel), online SVR(local kernel), C_1, C_2(penalty parameters), ε_1, ε_2 (ε-insensitive loss), p_1 , p_2 (kernel parameters), *train length* (the length of the online training dataset), *embedded dimension* (embedded dimension);

Output: prediction result *PredictL(i)*;
The detailed steps:

a. Select the local domain for the time series and achieve data preprocessing.
b. Batch SVR initial training with global kernel in the local domain.
c. For the update sample $X(k)$, the corresponding prediction result is $\tilde{X}_{\mathrm{Bg}}(k+1)$.
d. Compute the variance time series in the step above $\mathrm{var}(k) = X(k) - \tilde{X}_{\mathrm{Bg}}(k)$.
e. Online SVR training with local kernel function and prediction for the time series $\mathrm{var}(k)$ and the corresponding prediction result is $\tilde{\mathrm{var}}(k+1)$.
f. Online revised prediction value $\tilde{X}(k+1)$, $\tilde{X}(k+1) = \tilde{X}_{\mathrm{Bg}}(k+1) + \tilde{\mathrm{var}}(k+1)$.
g. For updating online time series (x_c, y_c), repeat step (c) to step (g).
h. When the update time series is beyond the local domain defined, update the local offline SVR and repeat step (c) to step (g).

3.6.3 Method III: Accelerated Decremental Fast Online SVR

With the increased size of online modeling data, the efficiency would decline. If the initial training dataset is large, it is difficult to obtain higher prediction efficiency, and there is an interaction between the precision and efficiency. In order

to obtain fast prediction, the online modeling data length should be cut shorter. Therefore, to improve the efficiency of the online SVR algorithm is to reduce the online sample set size.

The accelerated decremental fast online support vector regression (ADF-SVR) approach first selects nonsupport vectors as the decremental samples, and then, an accelerated decremental training is used to reduce the online training dataset. As a result, the efficiency of the method increases with less online dataset.

Generally, online SVR algorithm realizes the unlearning process by the decremental algorithm to forget or delete the furthest historical data sample of the S set and E set and R set. The decremental algorithms can be found in the paper [33, 38]. In this case, the computing will be heavy if the forgotten data sample is the support vector.

To decrease the computation and reduce the online dataset, we sample the data sample in the R set and S set, and a selective accelerated forgetting is adapted to improve the basic decremental algorithm. As a result, the online dataset could be cut and the computing complexity would be decreased and the operating efficiency would be improved. If more than one sample is ignored, the training algorithm is accelerated and effectively improved on the online.

If deleting a data sample x_d, removing it from the training set, the corresponding weight of the kernel function is θ_d, according to equation

$$\theta_d = -\sum_{i=1}^{n} \theta_i \tag{3.32}$$

If $x_d \in R$, and then,

$$\theta_d = 0 \tag{3.33}$$

Keeping θ_i, $i = 1 \ldots n$ unchanged, and correspondingly, by keeping the datasets S and R unchanged, the training ends.

If $x_d \in E$, and then,

$$|\theta_d| = C \tag{3.34}$$

Select the nearest data sample $x_{\text{sel}} \in E$, and $\theta_d + \theta_{\text{sel}} = 0$, change $x_{\text{sel}} \in R$; by keeping the other data sample unchanged, the training ends.

For the deleted samples, $x_{d1}, x_{d2}, \ldots$, corresponding weights $\theta_{d1}, \theta_{d2}, \ldots$, we can remove these samples from the training data set through remaining the support vector set unchanged. The other samples except support vector are the online training.

$$N = E \cup R = \{n_1, n_2, \ldots n_{d1-1}, n_{d1+1}, \ldots, n_{d2-1}, n_{d2+1}, \ldots, n_{l_N}\} \tag{3.35}$$

In the next incremental training, the amount of the matrix γ would be $l_N - 2$ and the online training of the computing complexity would decrease.

If the forgotten data sample is support vector, the support vector set would be

$$S = \{s_1, \ldots, s_{d1-1}, s_{d1+1}, \ldots, s_{d2-1}, s_{d2+1}, \ldots, s_{l_s}\} \tag{3.36}$$

In the next incremental training, the amount of the matrix β would be $l_s - 2$; correspondingly, the amount of the kernel matrix would be decreased and the computing would be reduced.

The new proposed algorithm that is ADF-SVR improves the efficiency by optimizing the decremental training strategy.

3.6.4 Method IV: Serial Segmental Online SVR

In order to improve the prediction precision for complicated time series, we present a novel segmental online SVR (SOSVR) algorithm for time series forecasting. Fast training speed is achieved by cutting the training dataset short. A segmental strategy is applied, and the online SVR model is stored by segments. The most suitable segmental model is selected to output the prediction value according to the matching degree between prediction neighborhood data and all the segmental models. As a result, the forecasting precision is improved.

The proposed SOSVR algorithm defines the segmentation condition, segmental principle (SGP), according to the update of the online data sample. If the SGP is met, cut the online SVR into little segment and store the segmental model $\mathrm{SOSVR}(\mathrm{sg}), \mathrm{sg} = 1, 2, 3\ldots$ as the historical model. The best segmental model selection criterion, select best predict principle (SBPP), is defined while prediction. If the segmental model $\mathrm{SOSVR}(\mathrm{sg}), \mathrm{sg} \in (1, 2, 3, \ldots)$ meets the SBPP, the prediction result is outputted by the best suitable model $\mathrm{SOSVR}(k)$. Given the online SVR local model, we apply the online training, segmenting and storing the segmental model, then update the output with the online data for model selection.

Because of the strategy of segmentation, the historical knowledge is kept by the series of segmental models. The best matching model is selected to output, and as a result, the prediction precision could be guaranteed under the condition of short online modeling dataset and high operating efficiency.

3.6.5 Method V: Multi-scale Parallel Online SVR

For complex nonlinear and nonstationary time series, they contain sequences long trend as well as strong neighborhood correlation. If a single fast online prediction model is used, the limit of sample size makes it difficult to obtain satisfactory results of prediction accuracy and efficiency because the algorithm could not take into account all relevant factors.

According to features of the online time series and considering training algorithm of online SVR for time series forecast, we can construct multiscale sub-time-sequences online. These sub-time-sequences could represent the complex properties of neighborhood and trend characteristics in various timescales and can be built by sampling with different sample rates. Then, these sub-time-series could be modeled with parallel online SVR algorithm to realize parallel fast online time series prediction with reduction in sample size [33, 35].

Therefore, the multi-scale reconstruction of time series with data sampling could reflect the characteristics of series long trends and short neighborhood nonlinear features. Data reconstruction can effectively reduce the size of the online datasets and preserve rich historical knowledge of samples. Independent parallel submodels could be obtained by modeling the sub-time-sequences with online SVR algorithm, respectively. Through the multi-scale reconstruction of the time series, the length of online modeling data is reduced. Meanwhile, the prediction efficiency could be improved and faster prediction can be achieved. Based on the above analysis, we proposed a multi-scale parallel online SVR algorithm (abbreviated as MSPO-SVR).

Therefore, the multi-scale reconstruction of time series with data sampling could reflect the characteristics of series long trends and short neighborhood nonlinear features. Data reconstruction can effectively reduce the size of the online datasets and, on the other hand, preserve the rich historical knowledge of samples. Independent parallel submodels could be obtained by modeling the sub-time-sequences with online SVR algorithm, respectively. Through the multi-scale reconstruction of the time series, the length of online modeling data is reduced. Meanwhile, the prediction efficiency could be improved and faster prediction can be achieved. According to the analysis above, this book presents an MSPO-SVR algorithm.

The main steps of MSPO-SVR algorithm are as follows:

a. Modeling stage: Firstly, the different timescale sub-time-series of neighborhood features and long trends by sampling with different sampling intervals to initial modeling data. Then, a series of timescale parallel online SVR models represented different timescale characteristics are obtained by training the sub-time-series using multiple parallel online SVR models.
b. Forecasting stage: With the online time series updating, the most suitable prediction model is selected by matching the online samples and every time-scale submodels. The output is obtained, and then, the multi-scale sub-time-series is updated, and at the same time, the corresponding submodels are training online, and so on.

The MSPO-SVR algorithm flowchart is shown as Fig. 3.1.

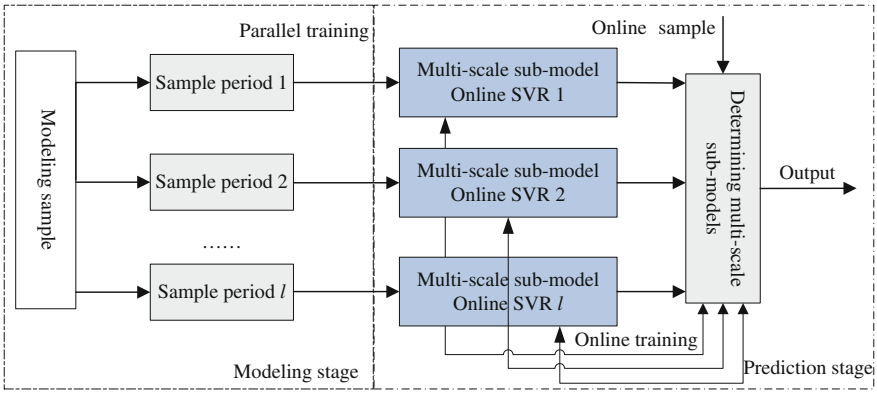


Fig. 3.1 Schematic of MSPO-SVR algorithm

3.7 Discussion on Optimized Kernel-Based Online SVR

3.7.1 Analysis and Comparison of Five Optimized Online SVR Algorithms

To compare the five proposed optimized online SVR algorithms and evaluate the adaptability of the different optimized online SVR algorithms, we compare the prediction error and operating time for varied time series datasets.

Define:

$$\eta_{\mathrm{MAE}} = \frac{\sum_{i}^{N} \mathrm{MAE}_i / N}{\sum_{i}^{N} \mathrm{MAE}_{\text{Online SVR}} / N} \tag{3.37}$$

Here, $\sum_i^N \mathrm{MAE}_i$ is the sum of prediction MAE for varied time series datasets of certain improved online SVR algorithm, and the $\sum_i^N \mathrm{MAE}_{\text{Online SVR}}$ is the one for the basic online SVR algorithm, N is the number of predicted time series, η_{MAE} represents the prediction precision of the proposed algorithm, the more the η_{MAE}, the worse is the precision.

Define:

$$\eta_{\mathrm{NRMSE}} = \frac{\sum_{i}^{N} \mathrm{NRMSE}_i}{\sum_{i}^{N} \mathrm{NRMSE}_{\text{Online SVR}}} \tag{3.38}$$

Here, $\sum_i^N \text{NRMSE}_i$ is the sum of prediction NRMSE for varied time series datasets of certain improved online SVR algorithm, and the $\sum_i^N \text{NRMSE}_{\text{Online SVR}}$ is the one for the basic online SVR algorithm, N is the number of predicted time series, η_{NRMSE} represents the prediction precision of the proposed algorithm, the more the η_{NRMSE}, the worse is the precision.

Define:

$$\eta_{\text{OpTime}} = \frac{\sum_i^N \text{OpTime}_i}{\sum_i^N \text{OpTime}_{\text{Online SVR}}} \tag{3.39}$$

Here, $\sum_i^N \text{OpTime}_i$ is the sum of operating time for varied time series datasets of certain improved online SVR algorithm, and the $\sum_i^N \text{OpTime}_{\text{OnlineSVR}}$ is the one for the basic online SVR algorithm, N is the number of predicted time series, η_{NRMSE} represents the prediction precision of the proposed algorithm, the more the η_{OpTime}, the worse is the efficiency.

From the Table 3.1, it can be seen that among the two types of kernel combination methods: KCO-SVR and VPCLO-SVR, the KCO-SVR has low efficiency than the basic online SVR, while the improved VPCLO-SVR can get very higher operating efficiency. Due to the kernel combining strategy, both types of methods can obtain more precise prediction results than the basic online SVR algorithm.

From the Table 3.2, the operating efficiency of three types of fast online SVR algorithms is higher than the basic online SVR. Compared to basic online SVR, the ADF-SVR can achieve the prediction efficiency improved 40–80 %, and the SOSVR can keep the efficiency with the basic online SVR. The prediction precision of the three types of methods increases gradually. The SOSVR can realize the prediction precision improved 5–10 %, while the MSPO-SVR can achieve the precision increasing more than 20 %.

Table 3.1 Evaluation of two types of kernel-combined online SVR algorithms for online time series prediction

Index	Algorithms	η_{MAE}	η_{NRMSE}	η_{OpTime}
1	KCO-SVR	0.904	0.774	1.891
2	VPCLO-SVR	0.920	0.789	0.306

Table 3.2 Evaluation of three types of sample-reduced online SVR algorithms

Index	Algorithms	η_{MAE}	η_{NRMSE}	η_{OpTime}
1	ADF-SVR (accelerated)	1.030	0.933	0.364
2	SOSVR	0.950	0.925	1.001
3	MSPO-SVR	0.642	0.861	2.230

It can be concluded from the evaluation experiments above that the proposed five types of improved online algorithms can meet the various applications. In the real applications, different algorithm can be selected according to the precision and operating efficiency to meet the different demands.

3.7.2 Application Example

To solve the problem with different application, an online adaptive data-driven fault prognosis and prediction strategy are presented as shown in Fig. 3.2.

The key problem for the algorithm application is how to choose the best models to do the prediction. In this book, online prognostic methods and models first need to consider actual conditions, such as application goal and system resource, prediction accuracy, and efficiency, and then, corresponding algorithm is chosen according to the characteristics of data. As a result, various forecasting demands can be implemented.

Fault prognostic model selection criterion is recommended as follows. (a) When the prediction accuracy is preferred, we recommend the kernel function–combined online SVR. The approach combines the excellent trend-fitting characteristic of global kernel function and powerful neighborhood nonlinear approximation ability of local kernel functions. This approach can achieve higher prediction accuracy compared with the single kernel function methods. (b) When the operation efficiency is focused, we introduce the accelerated online SVR algorithm, but the prediction accuracy is relatively worst. The serial SOSVR and the parallel multi-scale online SVR use the serial and parallel strategies to cut the dataset short; the former efficiency is higher than the latter. If the data reflect the time-domain multi-scale features, we suggest choosing the latter. (c) If the equipment or system resources, including computing, memory, are enough to support more complicated online algorithms, we can also use the strategies of

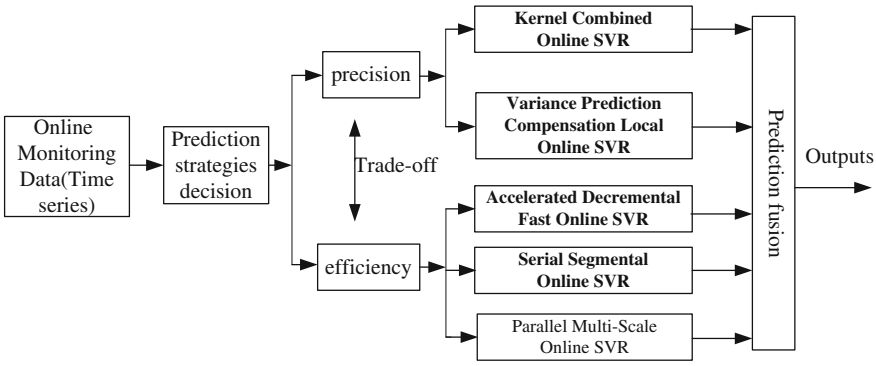


Fig. 3.2 Model of online fault prognostic strategies

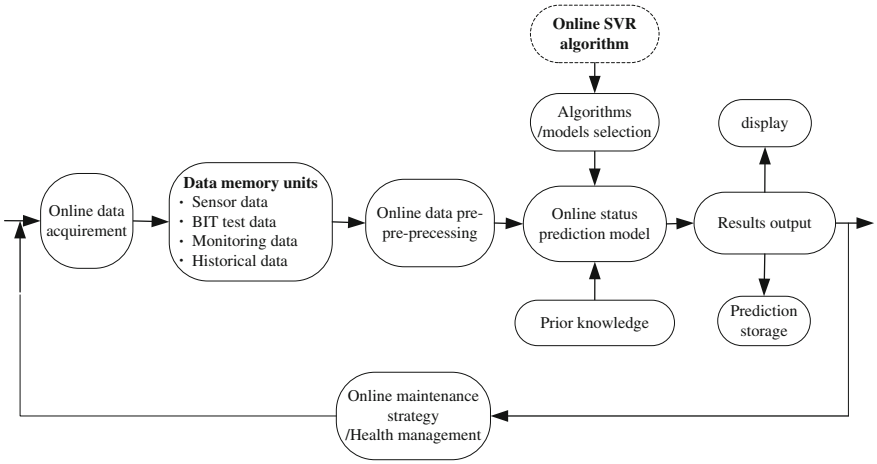


Fig. 3.3 Structure of online adaptive monitoring prediction system

model fusion with different online SVR algorithms. As a result, we can achieve complementary between the predicted results.

The online prognostic system based on adaptive online SVR algorithms is shown in Fig. 3.3.

Online data acquisition can be done with the data acquisition unit, monitoring unit, or embedded BIT unit of the system itself. These acquired data constructed the original online time series data samples for prediction. Online preprocessing is necessary to the raw data, such as data reduction, uncertainty management, component analysis etc.

Online status monitoring and prediction system can select the corresponding algorithm and model for different application demands. As well the online model could be selected according to some prior system knowledge. And if the system computing resource is enough, the prediction algorithm and model could be operated online and real time in the system-embedded CPU controller. In the end, the forecasting output can display locally or output to dedicated data interface with other subsystems. Meanwhile, according to forecast results, the operator can take appropriate maintenance strategy or health management.

References

1. Bashkirov OA, Braverman EM, Muchnik IB (1964) Potential function algorithms for pattern recognition learning machines. Automat Remote Contr 25:629–631
2. Aizerman MA, Braverman EM, Rozonoer LI (1964) Theoretical foundations of the potential function method in pattern recognition learning. Automat Remote Contr 25:821–837
3. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) Proceedings of 5th Annual ACM workshop computation learning theory, Pittsburgh, pp 144–152

4. Vapnik V (1982) Estimation of dependences based on empirical data. Springer, New York
5. Vapnik VN (1995) The nature of statistical learning theory. Springer, New York
6. Statistical Learning Theory (1998)(Adaptive and Learning Systems for Signal Processing, Communications and Control). Wiley, London
7. Cortes C, Vapnik VN (1995) Support vector networks. Mach Learn 20:273–297
8. Schölkopf B, Smola A, Müller K-R (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10:1299–1319
9. Mika S, Rätsch G, Weston J, Schölkopf B, Müller K-R (1999) Fisher discriminant analysis with kernels. In: Proceedings of IEEE neural networks for signal processing workshop, NNSP 99
10. Frieß TT, Cristianini N, Campbell C (1998) The Kernel-Adatron algorithm: a fast and simple learning procedure for support vector machines. Paper presented at the 15th International conference machine learning, San Mateo
11. Van Hulle MM (1998) Kernel-based equiprobabilistic topographic map formation. Neural Comput 10:1847–1871
12. Schölkopf B, Burges C, Smola A (eds) (1998) Advances in Kernel methods, support vector learning. MIT Press, Cambridge
13. Saha B, Goebel K, Poll S, Christophersen J (2007) An integrated approach to battery health monitoring using Bayesian regression and state estimation. In: Proceedings of autotestcon, pp 646–653
14. Zio E, Di Maio F (2010) A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. Reliab Eng Syst Saf 95:49–57
15. Luo M, Wang D, Pham M, Low CB, Zhang JB, Zhang DH, Zhao YZ (2005) Model-based fault diagnosis/prognosis for wheeled mobile robots: a review. In: Proceedings of IECON, pp 2267–2272
16. Basak D, Pal S, Patranabis DC (2003) Support vector regression. Neural Inf Process: Lett Rev 11(10):203–224
17. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222
18. Cauwenberghs G, Poggio T (2001) Incremental and decremental support vector machine learning. Mach Learn 44(13):409–415
19. Kim H, Park H (2004) Incremental and decremental least squares support vector machine and its application to drug design. In: Proceedings of the 2004 IEEE computational systems bioinformatics conference, Stanford, pp 656–657
20. Kivinen J, Smola AJ, Williamson RC (2004) Online learning with Kernels. IEEE Trans Signal Process 52(8):2165–2176
21. Ma JH, Theiler J, Perkins S (2003) Accurate online support vector regression. Neural Comput 15(11):2683–2703
22. Martin M (2002) Online support vector machine regression. Springer, Berlin. ECML, LNAI, pp 282–294
23. Syed NA, Liu H, Sung KK (1999) Incremental learning with support vector machines. In: Proceedings of international joint conference on artificial intelligence, vol 2. Stockholm, pp 272–276
24. Vijayakumar S, Ogawa H (1999) RKHS-based functional analysis for exact incremental learning. Neurocomputing 29(1/3):85–113
25. Engel Y, Mannor S, Meir R (2004) The Kernel recursive least- squares algorithm. IEEE Trans Signal Process 52(8):2275–2285
26. Jiao L, Bo L, Wang L (2007) Fast sparse approximation for least squares support vector machine. IEEE Trans Neural Netw 18(3):685–697
27. Wang DC, Jiang B (2007) Review of SVM-based control and online training algorithm. J Syst Simul 19(6):1177–1181
28. Wu Q, Liu WY, Yang YH (2007) Time series online prediction algorithm based on least squares support vector machine. J Cent Sou Univ Technol 14(3):442–446

29. Zhang HR, Wang XD (2006) Incremental and online learning algorithm for regression least squares support vector machine. Chin J Comput 29(3):400–406
30. Wikipedia Online. http://en.wikipedia.org/wiki
31. Tutorial slides by Andrew Moore. http://www.cs.cmu.edu/~awm
32. Vapnik V (1995) The nature of statistical learning theory. Springer, New York. ISBN 0-387-94559-8
33. Burges C (1998) A tutorial on support vector machines for pattern recognition. In Data mining and knowledge discovery, vol 2. Kluwer Academic Publishers, Boston
34. Vapnik V, Golowich S, Smola A (1997) Support vector method for function approximation, regression estimation, and signal processing. In: Mozer M, Jordan M, Petsche T (eds) Advances in neural information processing systems 9. MIT Press, Cambridge, pp 281–287
35. Brand M, Hertzmann A (2000) Style machines. In: Proceedings of ACM Siggraph00,pp 183–192
36. Bregler C, Covell M, Slaney M (1997) Video rewrite: driving visual speech with audio. In: Proceedings of ACM Siggraph97, pp 353–360
37. Ghahramani Z, Jordan M (1997) Factorial hidden Markov models. Machine Learn 29:245–273
38. Jojic N, Petrovic N, Frey BJ, Huang TS (2000) Transformed hidden Markov models: estimating mixture models of images and inferring spatial transformations in video sequences. In: Proceedings of IEEE conference on computer vision pattern recognition
39. Pullen K, Bregler C (2001) Life-like animations with motion textures. Available from <http://graphics.stanford.edu/~pullen/motion_texture/>
40. Pavlovic V, Regh JM (2000) Impact of dynamic model learning on classification of human motion. In: Proceedings of CVPR00
41. Hyvinen A, Karhunen J, Oja E (2001) Independent component analysis. Wiley, New York
42. Bartlett MS, Movellan JR, Sejnowski TJ (2002) Face recognition by independent component analysis. IEEE Trans Neural Netw 13(6):1450–1464
43. Liu C, Wechsler H (2003) Independent component analysis of Gabor features for face recognition. IEEE Trans Neural Netw 14(4):919–928
44. Yang J, Gao X, Zhang D, Yang J (2005) Kernel ICA: an alternative formulation and its application to face recognition. Pattern Recogn 38(10):1784–1787

Chapter 4
Kernel Principal Component Analysis (KPCA)-Based Face Recognition

4.1 Introduction

As a subfield of pattern recognition, face recognition (or face classification) has become a hot research point. In pattern recognition and in image processing, feature extraction based no dimensionality reduction plays the important role in the relative areas. Feature extraction simplifies the amount of resources required to describe a large set of data accurately for classification and clustering. On the algorithms, when the input data are too large to be processed and it is suspected to be notoriously redundant (much data, but not much information), then the input data will be transformed into a reduced representation set of features also named features vector with linear transformation or the nonlinear transformation. Transforming the input data into the set of features is called feature extraction. If the features extracted are carefully chosen, it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input data.

In the last decade years, many algorithms had been applied to face recognition, such as neural network (NN) [1–3], template matching [4], support vector machine (SVM) [5], nonnegative matrix factorization (NMF) [6], subspace analysis [7], elastic graph matching [3], principal component analysis (PCA) [3]. PCA is a classical dimensionality reduction method which has been applied in many applications, such as data visualization, image reconstruction, and biomedical study. It is a linear transformation that searches for an orthonormal basis of a low-dimensional subspace, so-called the principal components subspace, which explains the variability of the data as much as possible. PCA's utility and success stem from the simplicity of the method that calculates the eigenvectors and eigenvalues of the sample covariance matrix of the data set. Due to the complex variations of illumination, expression, angle of view and rotation, etc., it is difficult to describe the facial features through a single algorithm. Therefore, most of the current researches on face recognition focus on the recognition problems under restricted conditions. A common face recognition system consists of preprocessing, feature extraction/selection, and recognition. Among them, feature extraction

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_4,

is one of the key parts. While in some cases, a linear transformation is not suitable for capturing the nonlinear structures of the data, in order to represent the nonlinear structure, the kernel PCA (KPCA) has been formulated in a reproducing kernel hilbert space (RKHS) framework. In KPCA, the computational cost depends on the sample size. When the sample size is very large, it is impractical to compute the principal components via a direct eigenvalue decomposition.

In the past research works, there are many applications such as image denoising [8], stochastic complexity regularization [9], Sequence outlier detection [10], face detection [11], facial expressions [12], biometrics recognition [13]. Many algorithms are presented in the previous work. PCA and linear discriminant analysis [14] are the most popular dimensionality reduction for feature extraction. For many complicated feature extraction applications, recently, the nonlinear kernel-based dimensionality reduction method are applied into extend the linear method to develop kernel component analysis and kernel discriminant analysis [15, 16]. With kernel method in the practical application, all training samples must be saved and computed for feature extraction, which occurs the time-consuming and space-storing problems. In order to solve these problems, we present a novel feature extraction, namely refined kernel principal component analysis (RKPCA). With RKPCA, only a few of training samples are computed in the algorithm procedure.

4.2 Kernel Principal Component Analysis

4.2.1 Principal Component Analysis

Kernel principal component analysis (KPCA) is the extension of PCA as the linear feature extraction. The main idea of KPCA is to project the input data from the linear space into the nonlinear space and then implement PCA in the nonlinear feature space for feature extraction. By introducing the kernel trick, PCA is extended into KPCA algorithm. The detailed theoretical derivation is shown as follows:

$$C = \frac{1}{n}\sum_{i=1}^{n}(\Phi(x_i) - \overline{\Phi})(\Phi(x_i) - \overline{\Phi})^T \tag{4.1}$$

where $\overline{\Phi} = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)$, and let $\widetilde{C} = \frac{1}{n}\sum_{i=1}^{n}\Phi(x_i)\Phi(x_i)^T$ and $Q = [\Phi(x_1), \ldots, \Phi(x_n)]$, then $\widetilde{C} = \frac{1}{n}QQ^T$. According to $\widetilde{R} = Q^TQ$, with the kernel function, then

$$\widetilde{R}_{ij} = \Phi(x_i)^T\Phi(x_j) = \langle\Phi(x_i), \Phi(x_j)\rangle = k(x_i, x_j) \tag{4.2}$$

Compute the eigenvectors $u_1, u_2, \ldots, u_m$ according to the mth eigenvalue $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ of R, then $w_1, w_2, \ldots, w_m$ is calculated by

$$w_j = \frac{1}{\sqrt{\lambda_j}}Qu_j, \quad j = 1, 2, \ldots, m \tag{4.3}$$

Accordingly, $R = \widehat{R} - 1_n\widehat{R} - \widehat{R}1_n + 1_n\widehat{R}1_n$, where $(1_n)_{ij} = {}^1\!/_n$ $(i,j = 1,2,\ldots,n)$, then

$$y_j = w_j^T x = \frac{1}{\sqrt{\lambda_j}} u_j^T [k(x_1,x), k(x_2,x), \ldots, k(x_n,x)] \tag{4.4}$$

PCA-based feature extraction needs to store the $r \times m$ coefficient matrix W, where r is the number of principal components, and m is the number of training samples. KPCA-based feature extraction needs to store the original sample information owing to computing the kernel matrix with all training samples, which leads to a huge store space and a high computing consuming.

4.2.2 Kernel Discriminant Analysis

Kernel discriminant analysis is based on a conceptual transformation from the input space into a nonlinear high-dimensional feature space. Suppose that M training samples $\{x_1, x_2, \ldots, x_M\}$ with L class labels take values in an N-dimensional space $\mathbb{R}^N$, the data in $\mathbb{R}^N$ are mapped into a feature space F via the following nonlinear mapping, $\Phi : \mathbb{R}^N \to F, x \mapsto \Phi(x)$. Consequently in the feature space F Fisher criterion is defined by

$$J(V) = \frac{V^T S_B^\Phi V}{V^T S_T^\Phi V} \tag{4.5}$$

where V is the discriminant vector, and S_B^Φ and S_T^Φ are the between-classes scatter matrix and the total population scatter matrix, respectively. According to the Mercer kernel function theory, any solution V belongs to the span of all training pattern in $\mathbb{R}^N$. Hence, there exist coefficients $c_p (p = 1,2,\ldots,M)$ such that

$$V = \sum_{p=1}^{M} c_p \Phi(x_p) = \Psi\alpha \tag{4.6}$$

where $\Psi = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_M)]$ and $\alpha = [c_1, c_2, \ldots, c_M]^T$. Suppose the data are centered, Fisher criterion is transformed into

$$J(\alpha) = \frac{\alpha^T KGK\alpha}{\alpha^T KK\alpha} \tag{4.7}$$

where $G = \mathrm{diag}(G_1, G_2, \ldots, G_L)$, and G_i is an $n_i \times n_i$ matrix whose elements are $\frac{1}{n_i}$, and K is the kernel matrix calculated by a basic kernel $k(x,y)$. The criterion given in (4.12) attains its maximum for the orthonormal vectors. There are numerous algorithms to find this optimal subspace and an orthonormal basis for it.

4.2.3 Analysis on KPCA and KDA

In this section, we analyze the above kernel learning methods including KPCA and kernel discriminant analysis (KDA) as follows:

Firstly, on KPCA, this nonlinearity is firstly mapping the data into another space using a nonlinear map, and then, PCA is implemented using the mapped examples. The mapping and the space are determined implicitly by the choice of a kernel function which computes the dot product between two input examples mapped into feature space via kernel function. If kernel function is a positive definite kernel, then there exists a map into a dot product space. The space has the structure of a so-called RKHS. The inner products in feature space can be evaluated without computing the nonlinear mapping explicitly. This allows us to work with a very high-dimensional, possibly infinite-dimensional RKHS. If a positive definite kernel is specified, we need to know neither the nonlinear mapping nor feature space explicitly to perform KPCA since only inner products are used in the computations. Commonly used examples of such positive definite kernel functions are the polynomial kernel and Gaussian kernel, each of them implying a different map and RKHS. PCA-based feature extraction needs to store the $r \times m$ coefficient matrix, where r is the number of principal components, and m is the number of training samples. KPCA-based feature extraction needs to store the original sample information owing to computing the kernel matrix, which leads to a huge store and a high computing consuming. In order to solve the problem, we apply the LS-SVM to build the sparse KPCA.

Secondly, KDA is successfully to solve the nonlinear problem endured by linear discriminant analysis (LDA) as a traditional dimensionality reduction technique for feature extraction. In order to overcome this weakness of LDA, the kernel trick is used to represent the complicated nonlinear relationships of input data to develop KDA algorithm. Kernel-based nonlinear feature extraction techniques have attracted much attention in the areas of pattern recognition and machine learning. KDA has been applied in many real-world applications owing to its excellent performance on feature extraction.

Thirdly, both KDA and KPCA endure the kernel function and its parameters. kernel function and its parameter have significant influence on feature extraction owing to the fact that the geometrical structure of the data in the kernel mapping space is determined totally by the kernel function. If an inappropriate kernel is used, the data points in the feature space may become worse. However, choosing the kernel parameters from a set of discrete values will not change the geometrical structures of the data in the kernel mapping space.

So, it is feasible to improve the performance of KPCA with sparse analysis and kernel optimization. We reduce the training samples with sparse analysis and then optimize kernel structure with the reduced training samples.

4.3 Related Improved KPCA

4.3.1 Kernel Symmetrical Principal Component Analysis

Recently, Yang et al. [17] proposed a symmetrical principal component analysis (SPCA) algorithm according to the symmetry of faces for face recognition. SPCA algorithm utilizes efficiently the symmetry of facial images. But SPCA also has some disadvantages; for example, when the asymmetry increases, the performance of SPCA will degenerate rapidly. By integrating the advantages of kernel method with ones of SPCA algorithm, this book proposes a kernel-based SPCA (KSPCA) algorithm, which, based on theoretical analysis and experimental results, has better performance in comparison with SPCA and KPCA.

The idea of SPCA algorithm comes from odd–even decomposition problem. The main idea is the odd–even decomposition as follows:

The function $f(t)$ can be decomposed into an even function $f_e(t)$ and the odd function $f_o(t)$, i.e., $f(t) = f_e(t) + f_o(t)$. With the decomposition, $f_e(t)$ and $f_o(t)$ can be presented by the linear combination of a set of odd/even symmetrical basic functions. Also, any function can be composed of a set of even symmetrical basic functions and a set of odd symmetrical basic functions. In the practical applications, the sine and cosine functions are chosen as odd symmetrical basic function and even symmetrical basic function, that is the Fourier decomposition. In the face recognition, the symmetry is to be the horizontal mirror symmetry with the vertical midline of the image as its axis.

Suppose that x_k is one face images from one image set, and where $k = 1, 2,\ldots, N$, N denotes the number of all training face image. With the odd–even decomposition theory, x_k is decomposed as $x_k = x_{ek} + x_{ok}$, with $x_{ek} = (x_k + x_{mk})/2$ be the even symmetrical image and $x_{ok} = (x_k - x_{mk})/2$ denotes the odd symmetrical image where x_{mk} is the mirror image of x_k.

Suppose M, Me, Mo denote the covariance matrix of the set of face images x_k, $x_{ek} = (x_k + x_{mk})/2$ and $x_{ok} = (x_k - x_{mk})/2$, respectively, i.e., $M = 1/N \sum_{k=1}^{N} x_k x_k^T$, $Me = 1/N \sum_{k=1}^{N} x_{ek} x_{ek}^T$, $Mo = 1/N \sum_{k=1}^{N} x_{ok} x_{ok}^T$. It is evident that the eigenvalue decomposition on M is equivalent to the eigenvalue decomposition on Me, Mo. So the original image x_k can reconstructed linearly from the eigenvectors of Me, Mo. The researchers present its kernel version, called kernel symmetrical principal component analysis (KSPCA) as follows. The input space is mapped into the feature space with the nonlinear mapping. In the linear space, it is easy to decompose the original image x_k as $x_k = x_{ek} + x_{ok}$, with $x_{ek} = (x_k + x_{mk})/2$ be the even symmetrical image and $x_{ok} = (x_k - x_{mk})/2$ denotes the odd symmetrical image where x_{mk} is the mirror image of x_k. But in the nonlinear mapping space, it is difficult to decompose $\phi(x_k)$ both in feature space because the form of the nonlinear function $\phi(x_k)$ is not given with the math function. But it may be relatively simple to solve this problem when the nonlinear function $\phi(\cdot)$ within some function family such as Gaussian kernel, polynomial kernel functions. This book focuses on the

family of polynomial function if the nonlinear function $\phi(\cdot)$ maps the input space into feature space within the family of polynomial function.

Based on the principal component extraction, the test image $\phi(x)$ into the eigenvectors W_f in the feature space according to

$$(W_f \cdot \phi(x)) = \sum_{i=1}^{M} \alpha_i (\phi(x_i) \cdot \phi(x)) \tag{4.8}$$

Face image x into even symmetrical image $x_{ek} = (x_k + x_{mk})/2$ and the odd symmetrical image $x_{ok} = (x_k - x_{mk})/2$ on the nonlinear feature space. As we know, sine/cosine functions are used paired up in Fourier transformation for spectrum analysis, the odd/even principal components cannot be used in this way, because the odd/even symmetrical feature vectors show different sensitiveness to disturbs. For example, angle of view, rotation, and the unevenness of illumination can bring asymmetry to facial images. The asymmetry is embodied solely in the odd symmetrical vectors. Thus, the odd symmetrical vectors are more liable to the effect of these disturbs, while the even symmetrical vectors are more stable. The robustness of the algorithm will be weakened if we do not discriminate between the odd/even symmetrical vectors. In face recognition, the odd and even symmetrical principal components hold different energy. For the PIE problem faced by face recognition, the symmetry of faces overwhelms their asymmetry, though the asymmetry is quite valuable in some other fields such as the thermal imaging of faces. Thus, even symmetrical components will hold more energy than odd symmetrical components. This means even symmetrical components are more important than the odd symmetrical components. Of course, it does not mean that we should completely discard the odd symmetrical components, because some of them still contain important information for face recognition. Both the odd symmetrical components and the even symmetrical components are used, and the even symmetrical components should be reinforced while the odd symmetrical components are suppressed. For feature selection in KSPCA, we adopt the strategy similar to that of SPCA, i.e., order eigenvectors according to their energy or variance and then select eigenvectors with more energy or greater variance. Since the variance of the even symmetrical components is bigger than the variance of the correlative components, the variance of the correlative components is bigger than the variance of the odd symmetrical components. So it is natural to consider the even symmetrical components first, then the correlative components, and the odd symmetrical components if necessary.

4.3.2 Iterative Kernel Principal Component Analysis

As mentioned in the previous section, it is impractical to compute the principal components via a direct eigenvalue decomposition while we have a large sample size in KPCA. Besides, it is also recognized that KPCA is sensitive to outliers.

To overcome these two problems, we propose an iterative KPCA method, which advances iterative updating techniques for robust estimation of principal directions. We note that although the number of training instances determines the dimensionality of the kernel in KPCA, an incremental (or online) setting cannot guarantee asymptotic convergence of the iterative KPCA. In our proposed iterative RKPCA, we assume that there are only finite main features of these observations. That is, we can choose an arbitrary fixed basis of the kernel so that the size of the kernel is fixed as in KHA. Thus, the main issue is to find a way to choose the number of principal components to compress the data size and to prove convergence of our proposed method.

4.4 Adaptive Sparse Kernel Principal Component Analysis

In this section, we present a novel learning called refined kernel principal component analysis (RKPCA) with the viewpoint of support vector machine (SVM). In SVM, only few support vectors are meaning for classification, and other samples can be ignored for training the classifier. We introduce the idea of SVM into KPCA and choose the few training samples for KPCA. Firstly, we apply a least squares support vector machine (LS-SVM) formulation to KPCA which is interpreted as a one-class modeling problem with a target value equal to zero around which one maximizes the variance. Then, the objective function can be described as

$$\max_{w} \sum_{i=1}^{N} \left[0 - w^T\left(\phi(x_i) - u^{\phi}\right)\right]^2 \tag{4.9}$$

where $\phi : \mathbb{R}^N \to \mathbb{R}^l$ denotes the mapping to a high-dimensional feature space and $u^{\phi} = \frac{1}{N}\sum_{i=1}^{N} \phi(x_i)$. We formulate KPCA with direct sparse kernel learning method, and we also use the phase "expansion coefficients" and "expansion vectors." Suppose a matrix $Z = [z_1, z_2, \ldots, z_{N_z}]$, $Z \in \mathbb{R}^{N \times N_z}$, composed of N_z expansion vectors, and β_i $(i = 1, 2, \ldots, N_z)$ $(N_z < N)$ are expansion coefficients, we modify the optimization problem to the following constraint optimization problem:

$$\begin{aligned} &\max_{w,e} J(w, e) = -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ &\text{subject to } e_i = w^T\left(\phi(x_i) - u^{\phi}\right), \quad i = 1, 2, \ldots, N \\ &w = \sum_{i=1}^{N_z} \phi(z_i)\beta_i \end{aligned} \tag{4.10}$$

where $\phi(Z) = \left[\phi(z_1), \phi(z_2), \ldots, \phi(z_{N_z})\right]$. Now our goal is to solve the above optimization problem. We can divide the above optimization problem into two steps, one is to find the optimal expansion vectors and expansion coefficients;

second is to find the optimal projection matrix. When Z is fixed, then we apply the kernel function, that is, $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$. Given a random Z, then the above problem is same to the following problem.

$$W(Z) := \max_{\beta, e} -\frac{1}{2}\beta^T K_z \beta + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \quad \text{subject to } e_i = \beta^T g(x_i), \quad i = 1, 2, \ldots, N \tag{4.11}$$

where $g(x_i) = \left[k(z_1, x_i) - \frac{1}{N}\sum_{q=1}^{N} k(z_1, x_q) \cdots k(z_{N_z}, x_i) - \frac{1}{N}\sum_{q=1}^{N} k(z_{N_z}, x_q)\right]^T$, $\beta = [\beta_1, \beta_2, \ldots, \beta_{N_z}]^T$, $K_z = [k(z_i, z_j)]$. The solution of the above-constrained optimization problem can often be found by using the so-called Lagrangian method. We define the Lagrangian method

$$L(\beta, e, \alpha) = -\frac{1}{2}\beta^T K_z \beta + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i \left(e_i - \beta^T g(x_i)\right) \tag{4.12}$$

with the parameter α_i, $i = 1, 2, \ldots, N$. The Lagrangian L must be maximized with respect to β, α_i, and e_i $i = 1, 2, \ldots, N$, and the derivatives of L with respect to them must vanish. We can obtain the optimal solution α^z, which is an eigenvector of the $G^T (K_z)^{-1} G$ corresponding to the largest eigenvalue.

$$\beta^z = (K_z)^{-1} G \alpha^z \tag{4.13}$$

where $G = [g(x_1), g(x_2), \ldots, g(x_N)]$, and now our goal is to find the optimal Z that maximizes the following equation:

$$W(Z) = -\frac{1}{2}(\beta^z)^T K_z (\beta^z) + \frac{\gamma}{2}(\beta^z)^T G G^T (\beta^z) \tag{4.14}$$

So it is easy to achieve Z^* to maximize the above Eq. (4.10). After we obtain Z^*, we can obtain $A = [\alpha_1, \alpha_2, \ldots, \alpha_m]$ corresponding to the largest eigenvalue of $G^T (K_z)^{-1} G$. Then, we can obtain

$$B = (K_z)^{-1} G A \tag{4.15}$$

Then, for a input vector x, its feature Y_x is calculated with the following equation:

$$Y_x = B K_{zx} \tag{4.16}$$

where K_{zx} is the kernel vector calculated with the input vector x and the refined training set Z^*.

As above discussion from the theoretical viewpoints, sparse kernel principal component analysis (SKPCA) algorithm adaptively chooses the few samples from the training sample set but little influence on recognition performance, which saves much space of storing training samples for computing the kernel matrix with lower time-consuming. So in the practical applications, SKPCA can solve the limitation from KPCA owing to its high store space and time-consuming its ability on feature extraction.

So from the theory viewpoint, SKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

In this section, we present a novel learning called sparse data-dependent kernel principal component analysis (SDKPCA) with the viewpoint of LS-SVM to solve the following problem. That is, the first is that all training samples need to be stored for computing the kernel matrix during kernel learning and the second is that the kernel and its parameter have the heavy influence on performance of kernel learning. We reduce the training samples with sparse analysis and then optimize kernel structure with the reduced training samples.

4.4.1 Reducing the Training Samples with Sparse Analysis

Firstly, we apply a LS-SVM machine formulation to KPCA which is interpreted as one-class modeling problem with a target value equal to zero around which one maximizes the variance. Secondly, we introduce data-dependent kernel into SKPCA, where the structure of the input data is adaptively changed regard to the distribution of input data. Then, the objective function can be described as

$$\max_{w} \sum_{i=1}^{N} \left[0 - w^T\left(\phi(x_i) - u^{\phi}\right)\right]^2 \tag{4.17}$$

where $\phi : \mathbb{R}^N \to \mathbb{R}^l$ denotes the mapping to a high-dimensional feature space and $u^{\phi} = \left(1/N\right) \sum_{i=1}^{N} \phi(x_i)$. The interpretation of the problem leads to the following optimization problem:

$$\begin{aligned} \max_{w,e} J(w,e) &= -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ \text{subject to } e_i &= w^T\left(\phi(x_i) - u^{\phi}\right), \quad i = 1, 2, \ldots, N \end{aligned} \tag{4.18}$$

We also apply the direct sparse kernel learning method to KPCA. Here, we also use the phase "expansion coefficients" and "expansion vectors." Suppose a matrix $Z = [z_1, z_2, \ldots, z_{N_z}]$, $Z \in \mathbb{R}^{N \times N_z}$, composed of N_z expansion vectors, and β_i $(i = 1, 2, \ldots, N_z)$ $(N_z < N)$ are expansion coefficients, we modify the optimization problem to the following problem:

$$\begin{aligned} \max_{w,e} J(w,e) &= -\frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \\ \text{subject to } e_i &= w^T\left(\phi(x_i) - u^{\phi}\right), \quad i = 1, 2, \ldots, N \\ w &= \sum_{i=1}^{N_z} \phi(z_i)\beta_i \end{aligned} \tag{4.19}$$

where $\phi(Z) = \left[\phi(z_1), \phi(z_2), \ldots, \phi(z_{N_z})\right]$. Now our goal is to solve the above optimization problem. We divide the above optimization problem into two steps, one is to find the optimal expansion vectors and expansion coefficients; second is to find the optimal projection matrix. Firstly, we reduce the above optimization problem, and then, we can obtain

$$\begin{aligned} \max_{Z,\beta,e} J(Z,\beta,e) &= -\frac{1}{2}\left(\sum_{r=1}^{N_z}\phi(z_r)\beta_r\right)^T\left(\sum_{s=1}^{N_z}\phi(z_s)\beta_s\right) + \frac{\gamma}{2}\sum_{i=1}^{N}e_i^2 \\ \text{subject to } e_i &= \left(\sum_{r=1}^{N_z}\phi(z_r)\beta_r\right)^T\left(\phi(x_i) - u^{\phi}\right), \quad i = 1,2,\ldots,N \end{aligned} \tag{4.20}$$

where Z is variable. When Z is fixed, then

$$\begin{aligned} \max_{\beta,e} J(\beta,e) &= -\frac{1}{2}\sum_{r=1}^{N_z}\sum_{s=1}^{N_z}\beta_s\beta_r\phi(z_r)^T\phi(z_s) + \frac{\gamma}{2}\sum_{i=1}^{N}e_i^2 \\ \text{subject to } e_i &= \left(\sum_{r=1}^{N_z}\beta_r\phi(z_r)^T\right)\left(\phi(x_i) - u^{\phi}\right), \quad i = 1,2,\ldots,N \end{aligned} \tag{4.21}$$

We apply the kernel function, that is, $k(x,y) = \langle\Phi(x), \Phi(y)\rangle$, given a random Z, and then, the above problem is same to the following problem.

$$\begin{aligned} W(Z) &:= \max_{\beta,e} -\frac{1}{2}\beta^T K_z\beta + \frac{\gamma}{2}\sum_{i=1}^{N}e_i^2 \\ \text{subject to } e_i &= \beta^T g(x_i), \quad i = 1,2,\ldots,N \end{aligned} \tag{4.22}$$

where $\beta = \left[\beta_1, \beta_2, \ldots, \beta_{N_z}\right]^T$, $K_{ij}^z = k(z_i, z_j)$ and $g(x_i) = \left[k(z_1,x_i) - \frac{1}{N}\sum_{q=1}^{N} k(z_1,x_q) \cdots k(z_{N_z},x_i) - \frac{1}{N}\sum_{q=1}^{N}k(z_{N_z},x_q)\right]^T$.

4.4.2 Solving the Optimal Projection Matrix

After the optimal solution of data-dependent kernel is solved, the optimal kernel structure is achieved which is robust to the changing of the input data. After this step, the next step is to solve the equation to obtain the optimized sparse training samples with the so-called Lagrangian method. We define the Lagrangian as

$$L(\beta,e,\alpha) = -\frac{1}{2}\beta^T K_z\beta + \frac{\gamma}{2}\sum_{i=1}^{N}e_i^2 - \sum_{i=1}^{N}\alpha_i\left(e_i - \beta^T g(x_i)\right) \tag{4.23}$$

with the parameter α_i, $i = 1, 2, \ldots, N$. The Lagrangian L must be maximized with respect to β, α_i, and e_i $i = 1, 2, \ldots, N$, and the derivatives of L with respect to them must vanish, that is,

$$\begin{cases} \frac{\partial L}{\partial \beta} = 0 \rightarrow K_z\beta = \sum_{i=1}^{N} \alpha_i g(x_i) \\ \frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i \\ \frac{\partial L}{\partial \alpha_i} = 0 \rightarrow e_i - \beta^T g(x_i) = 0 \end{cases} \tag{4.24}$$

Let $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ $(\alpha_{N\times 1})$, and $G = [g(x_1), g(x_2), \ldots, g(x_N)]$ $(G_{N_z\times N})$ and $E = [e_1, e_2, \ldots, e_N]^T$ $(E_{N\times 1})$, we can obtain

$$\begin{cases} K_z\beta = G\alpha \\ \alpha = \gamma E \\ E = G^T\beta \end{cases} \tag{4.25}$$

So, we can obtain $\beta = (K_z)^{-1}G\alpha$, then $E = G^T(K_z)^{-1}G\alpha$. It is easy to obtain the optimal solution α^z, which is an eigenvector of the $G^T(K_z)^{-1}G$ corresponding to the largest eigenvalue $\beta^z = (K_z)^{-1}G\alpha^z$. $W(Z)$ reaches the largest value when α^z is the eigenvector of $G^T(K_z)^{-1}G$ corresponding to the largest value, and $\beta^z = (K_z)^{-1}G\alpha^z$. The proof is described as follows:

Theorem 1 *$W(Z)$ reaches the largest value when α^z is the eigenvector of $G^T(K_z)^{-1}G$ corresponding to the largest value, and $\beta^z = (K_z)^{-1}G\alpha^z$.*

Proof Firstly, let us reconsider the Eq. (4.24) as follows:

When $\lambda > 0$, then

$$\begin{aligned} -\frac{1}{2}\beta^T K_z\beta &= -\frac{1}{2}\left[\alpha^T G^T\left((K_z)^{-1}\right)^T\right]K_z\left[(K_z)^{-1}G\alpha\right] \\ &= -\frac{1}{2}\left[\alpha^T G^T(K_z)^{-1}G\alpha\right] \\ &= -\frac{1}{2}\lambda\alpha^T\alpha \end{aligned} \tag{4.26}$$

Moreover, since $E = G^T(K_z)^{-1}G\alpha$, $G^T(K_z)^{-1}G\alpha = \lambda\alpha$, and $E = \lambda\alpha$, we obtain

$$\frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 = \frac{\gamma}{2}E^T E = \frac{\gamma}{2}\lambda^2\alpha^T\alpha \tag{4.27}$$

Since $\alpha^T\alpha = 1$, we can obtain

$$J(\beta, e) = -\frac{1}{2}\beta^T K_z\beta + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 = -\frac{1}{2}\lambda\alpha^T\alpha + \frac{\gamma}{2}\lambda^2\alpha^T\alpha = \frac{1}{2}\lambda^2\left(\gamma - \frac{1}{\lambda}\right) \tag{4.28}$$

From the above equation, we can see that $J(\beta, e)$ reaches the largest value when λ reaches the largest value.

From the above equation, we can see that $J(\beta, e)$ reaches the largest value when λ reaches the largest value. Now our goal is to find the optimal Z that maximizes $W(Z) = -\frac{1}{2}(\beta^z)^T K_z(\beta^z) + \frac{\gamma}{2}(\beta^z)^T GG^T(\beta^z)$.

After we obtain Z^*, and then compute the eigenvector $A = [\alpha_1, \alpha_2, \ldots, \alpha_m]$ of $G^T(K_z)^{-1}G$ corresponding to the following eigenproblem $G^T(K_z)^{-1}G\alpha = \lambda\alpha.$, then

$$B = (K_z)^{-1}GA. \tag{4.29}$$

4.4.3 Optimizing Kernel Structure with the Reduced Training Samples

For given kernel, we introduce that the data-dependent kernel with a general geometrical structure can obtain the different kernel structure with different combination parameters, and the parameters are self-optimized under the criterions. Data-dependent kernel $k'(x, y)$ is described as

$$k'(x, y) = f(x)f(y)k(x, y) \tag{4.30}$$

where $f(x)$ is a positive real-valued function x, and $k(x, y)$ is a basic kernel, e.g., polynomial kernel and Gaussian kernel. Amari and Wu [16] expanded the spatial resolution in the margin of a SVM by using $f(x) = \sum_{i \in SV} a_i e^{-\delta \left\| x - \widetilde{x_i} \right\|^2}$, where $\widetilde{x_i}$ is the ith support vector, and SV is a set of support vector, and a_i is a positive number representing the contribution of $\widetilde{x_i}$, and δ is a free parameter. We generalize Amari and Wu's method as

$$f(x) = b_0 + \sum_{n=1}^{N_z} b_n e(x, \widetilde{x}_n) \tag{4.31}$$

where $e(x, \widetilde{x}_n) = e^{-\delta \left\| x - \widetilde{x_n} \right\|^2}$, and δ is a free parameter, and $\widetilde{x}_n$ are called the "expansion vectors (XVs)," and N_z is the number of XVs, and b_n are the "expansion coefficients" associated with $\widetilde{x}_n$. The definition of the data-dependent kernel shows that the geometrical structure of the data in the kernel mapping space is determined by the expansion coefficients with the determinative XVs and free parameter. The objective function to find the adaptive expansion coefficients varied with the input data for the quasiconformal kernel. Given the free parameter δ and the expansion vectors $\{\widetilde{x}_i\}_{i=1,2,\ldots,N_Z}$, we create the matrix

$$E = \begin{bmatrix} 1 & e(x_1, \widetilde{x}_1) & \cdots & e(x_1, \widetilde{x}_{N_z}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e(x_M, \widetilde{x}_1) & \ldots & e(x_M, \widetilde{x}_{N_z}) \end{bmatrix} \tag{4.32}$$

Let $\xi = [b_0, b_1, b_2, \ldots, b_{N_z}]^T$ and $\Lambda = \mathrm{diag}(f(x_1), f(x_2), \ldots, f(x_M))$, the following equation is obtained

$$\Lambda 1_M = E\xi \tag{4.33}$$

where 1_M is a M-dimensional vector whose entries equal to unity. The expansion coefficient vector ξ is solved through optimizing an objective function designed for measuring the class separability of data in feature space with Fisher criterion and maximum margin criterion in our previous work [17]. We apply maximum margin criterion to optimized kernel as follows. The main idea of optimizing kernel structure with the reduced training samples is to find the optimal data-dependent kernel parameter vector ξ through optimizing an objective function. The kernel optimization algorithm procedure is described as follows:

$$\begin{aligned} &\max \quad J_{\mathrm{Fisher}} \\ &\text{subject to } \xi^T \xi - 1 = 0 \end{aligned} \tag{4.34}$$

where $J_{\mathrm{Fisher}} = \frac{\xi^T E^T B_0 E \xi}{\xi^T E^T W_0 E \xi}$, and let $J_1(\xi) = \xi^T E^T B_0 E \xi$ and $J_2(\xi) = \xi^T E^T W_0 E \xi$, then

$$\frac{\partial J_{\mathrm{Fisher}}(\xi)}{\partial \xi} = \frac{2}{J_2^2}(J_2 E^T B_0 E - J_1 E^T W_0 E)\xi \tag{4.35}$$

The optimal solution is achieved with the iteration method, and then, the optimal ξ is solved as follows:

$$\xi^{(n+1)} = \xi^{(n)} + \varepsilon \left(\frac{1}{J_2} E^T B_0 E - \frac{J_{\mathrm{Fisher}}}{J_2} E^T W_0 E \right) \xi^{(n)} \tag{4.36}$$

where ε is the learning rate with its definition of $\varepsilon(n) = \varepsilon_0(1 - \frac{n}{N})$, where ε_0 is the initialized learning rate, n and N are the current iteration number and the total iteration number, respectively.

4.4.4 Algorithm Procedure

For a set of training sample set, first, we optimize the kernel function $k'(x, y)$ with the given basic kernel function $k(x, y)$ and then implement SKPCA.

$$y = B^T V_{zx} \tag{4.37}$$

where $g(z_i,x)=k'(z_i,x)-\frac{1}{N}\sum_{q=1}^{N}k'(z_i,x_q)$, $V_{zx}=[\,g(z_1,x)\quad g(z_2,x)\ldots g(z_{N_z},x)]^T$. Since $w=\sum_{i=1}^{N_z}\phi(z_i)\beta_i^z$, so

$$y=\sum_{i=1}^{N_z}\beta_i^z\left[\phi(z_i)^T\left(\phi(x)-u^{\phi}\right)\right] \tag{4.38}$$

Let $\beta_z=\left[\beta_1^z\quad\beta_2^z\quad\cdots\quad\beta_{N_z}^z\right]^T$. For we choose m eigenvector α corresponding to m largest eigenvalue. Let $P=\left[\left(\beta_z^T\right)_1\quad\left(\beta_z^T\right)_2\quad\cdots\quad\left(\beta_z^T\right)_m\right]^T$, the feature can be obtained as follows:

$$z=PK_{zx} \tag{4.39}$$

As above discussion from the theoretical viewpoints, sparse data-dependent kernel principal component analysis (SDKPCA) chooses adaptively a few of samples from the training sample set, but a little influence on recognition performance, which saves much space of storing training samples on computing the kernel matrix with the lower time-consuming. So in the practical applications, SDKPCA can solve the limitation from KPCA owing to its high store space and time-consuming its ability on feature extraction. So from the theory viewpoint, SDKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

4.5 Discriminant Parallel KPCA-Based Feature Fusion

4.5.1 Motivation

How to perform a wonderful classification based on the multiple features becomes a crucial problem for pattern classification problem when multiple features are considered. As a very efficient method, data fusion is applied to solve it, which has been widely applied in many areas [1–3]. Existing fusion methods can be divided into the following three schemes: the first scheme is to integrate all assimilated multiple features into a final decision directly; the second is to combine the individual decisions made by every feature into a global final decision; and the third is to fuse the multiple features to one new feature for classification. In this book, we devote our attention to the third fusion scheme, i.e., feature fusion. Recently, many feature fusion methods for pattern classification were proposed in the lectures [4–6]. In this book, we focus on the linear combination fusion, but pay more attention to how to find the fusion coefficients, and propose so-called

discriminant feature fusion for supervising learning. The proposed discriminant fusion strategy has two advantages: (1) fused data have the largest class discriminant owing to obtaining the fusion coefficients by solving a constrained optimization problem created in the average margin criterion; (2) fusion coefficients are unique owing to they are equal to the elements of the eigenvector of one eigenvalue problem transformed by the above optimization problem. The advantage of algorithm lies in the following: (1) A constrained optimization problem based on maximum margin criterion is created to solve the optimal fusion coefficients, which causes that fused data have the largest class discriminant in the fused feature space. (2) An unique solution of optimization problem is transformed to an eigenvalue problem, which causes the proposed fusion strategy to perform a consistent performance. Besides the detailed theory derivation, many experimental evaluations also are presented.

4.5.2 Method

In this section, firstly, we introduce the basic idea of discriminant parallel feature fusion briefly and then emphasize the theory derivation of seeking the fusion coefficients in detailed.

Given a sample set $\{x_{ij}\}$ $(i = 1, 2, \ldots, C;\ j = 1, 2, \ldots, n_i)$ and multiple feature sets $\left\{y_{ij}^m\right\}$ $(i = 1, 2, \ldots, C;\ j = 1, 2, \ldots, n_i,\ m = 1, 2, \ldots, M)$, where M denotes the number of multiple features sets, the fused feature with the linear combination can be described as follows:

$$z_i^j = \sum_{m=1}^{M} a_m y_{ij}^m \tag{4.40}$$

where $a_m (m = 1, 2, \ldots, M)$ and z_{ij} $(i = 1, 2, \ldots, C,\ j = 1, 2, \ldots, n_i)$ denote the combination fusion coefficients and the fused feature, respectively.

Now we focus how to obtain a_m $(m = 1, 2, \ldots, M)$, and our goal is to find such fusion coefficients that they are unique and cause the largest class discriminant in the fused feature space. For supervised learning, we can calculate the average margin distance between two classes C_p^ℓ and C_q^ℓ in fused feature space $\mathbb{R}^\ell$ consisted of the fused feature $z_i^j = y_i^j \alpha$ $(i = 1, 2, \ldots, C,\ j = 1, 2, \ldots, n_i)$, where $\alpha = [a_1, a_2, \ldots, a_M]^T$ and $y_i^j = \left[y_{ij}^1, y_{ij}^2, \ldots, y_{ij}^M\right]$. The average margin distance can be defined by

$$\mathrm{Dis} = \frac{1}{2n} \sum_{p=1}^{L} \sum_{q=1}^{L} n_p n_q d\left(C_p^\ell, C_q^\ell\right) \tag{4.41}$$

where $d\left(C_p^\ell, C_q^\ell\right)$ denotes the margin distance between pth and qth classes. Given the feature vector z_i^j in the dimension-reduced space F^ℓ, and m_i^ℓ $(i = 1, 2, \ldots, L)$ and m^ℓ denote the mean of every class and the mean of total samples, respectively. Firstly, we can calculate $d\left(C_p^\ell, C_q^\ell\right)$ $(p = 1, 2, \ldots, L,\ q = 1, 2, \ldots, L)$ as follows:

$$d\left(C_p^\ell, C_q^\ell\right) = d\left(m_p^\ell, m_q^\ell\right) - S\left(C_p^\ell\right) - S\left(C_q^\ell\right) \tag{4.42}$$

where $S\left(C_p^\ell\right)$ $(p = 1, 2, \ldots, L)$ is the measure of the scatter of the class C_p^ℓ and $d\left(m_p^\ell, m_q^\ell\right)$ is the distance between the means of two classes. Let S_p^ℓ $(p = 1, 2, \ldots, L)$ denote the within-class scatter matrix of class p, then $\mathrm{tr}\left(S_p^\ell\right)$ $(p = 1, 2, \ldots, L)$ measures the scatter of the class p can be defined as follows:

$$\mathrm{tr}(S_p^\ell) = \frac{1}{n_p} \sum_{j=1}^{n_p} \left(z_p^j - m_p^\ell\right)^T \left(z_p^j - m_p^\ell\right) \tag{4.43}$$

And we can define that $\mathrm{tr}(S_B^\ell)$ and $\mathrm{tr}(S_W^\ell)$ denote the trace of between-classes scatter matrix and within-classes scatter matrix of dimension-reduced space F^ℓ, respectively, as follows:

$$\mathrm{tr}(S_B^\ell) = \sum_{p=1}^{L} n_p \left(m_p^\ell - m^\ell\right)^T \left(m_p^\ell - m^\ell\right) \tag{4.44}$$

$$\mathrm{tr}(S_W^\ell) = \sum_{p=1}^{L} \sum_{j=1}^{n_i} \left(z_p^j - m_p^\ell\right)^T \left(z_p^j - m_p^\ell\right) \tag{4.45}$$

Hence, $S\left(C_p^\ell\right) = \mathrm{tr}\left(S_p^\ell\right)$. So

$$\begin{aligned} Dis &= \frac{1}{2n} \sum_{p=1}^{L} \sum_{q=1}^{L} n_p n_q \left[d\left(m_p^\ell, m_q^\ell\right) - S\left(C_p^\ell\right) - S\left(C_q^\ell\right)\right] \\ &= \frac{1}{2n} \sum_{p=1}^{L} \sum_{q=1}^{L} n_p n_q d\left(m_p^\ell, m_q^\ell\right) - \frac{1}{2n} \sum_{p=1}^{L} \sum_{q=1}^{L} n_p n_q \left[\mathrm{tr}(S_p^\ell) + \mathrm{tr}(S_q^\ell)\right] \end{aligned} \tag{4.46}$$

Firstly, we use Euclidean distance to calculate $d\left(m_p^\ell, m_q^\ell\right)$ as follows:

$$\frac{1}{2n} \sum_{p=1}^{L} \sum_{q=1}^{L} n_p n_q d\left(m_p^\ell, m_q^\ell\right) = \frac{1}{2n} \sum_{p=1}^{L} \sum_{q=1}^{L} n_p n_q \left(m_p^\ell - m_q^\ell\right)^T \left(m_p^\ell - m_q^\ell\right) \tag{4.47}$$

According to Eqs. (4.5), (4.6), (4.8), and (4.9), it is easy to obtain

$$\frac{1}{2n}\sum_{p=1}^{L}\sum_{q=1}^{L} n_p n_q d\left(m_p^{\ell}, m_q^{\ell}\right) = \mathrm{tr}(S_B^{\ell}) \tag{4.48}$$

$$\frac{1}{2n}\sum_{p=1}^{L}\sum_{q=1}^{L} n_p n_q \left[\mathrm{tr}(S_p^{\ell})\right] = \frac{1}{2}\mathrm{tr}(S_W^{\ell}) \tag{4.49}$$

Hence, $\frac{1}{2n}\sum_{p=1}^{L}\sum_{q=1}^{L} n_p n_q \left[\mathrm{tr}(S_p^{\ell}) + \mathrm{tr}(S_q^{\ell})\right] = \mathrm{tr}(S_W^{\ell})$. We can obtain $Dis = \mathrm{tr}(S_B^{\ell}) - \mathrm{tr}(S_W^{\ell})$

In the previous work in [7], Li applied the maximum margin criterion to feature extraction by maximizing the average margin distance. We expect to create an optimization problem based on maximum margin criterion to seek an optimal projection vector α.

Proposition 4.1 *Let*

$$G = 2\sum_{i=1}^{L}\frac{1}{n_i}\left[\sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\left(y_i^j\right)^T y_i^k\right] - \sum_{i=1}^{L}\sum_{p=1}^{L}\sum_{j=1}^{n_i}\sum_{q=1}^{n_p}\left(\frac{1}{n}\left(y_i^j\right)^T y_p^q\right) - \sum_{i=1}^{L}\sum_{j=1}^{n_i}\left(y_i^j\right)^T y_i^j$$

Then, $Dis = \alpha^T G \alpha$.

Proof We can obtain

$$\mathrm{tr}(S_B) = \sum_{i=1}^{L}\frac{1}{n_i}\left[\sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\left(z_i^j\right)^T z_i^k\right] - \sum_{i=1}^{L}\sum_{p=1}^{L}\sum_{j=1}^{n_i}\sum_{q=1}^{n_p}\left(\frac{1}{n}\left(z_i^j\right)^T z_p^q\right) \tag{4.50}$$

$$\mathrm{tr}(S_W) = \sum_{i=1}^{L}\sum_{j=1}^{n_i}\left(z_i^j\right)^T z_i^j - \sum_{i=1}^{L}\frac{1}{n_i}\left[\sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\left(z_i^j\right)^T z_i^k\right] \tag{4.51}$$

From Eq. (4.2) $z_i^j = y_i^j \alpha$, we can obtain

Let $G = 2\sum_{i=1}^{L}\frac{1}{n_i}\left[\sum_{j=1}^{n_i}\sum_{k=1}^{n_i}\left(y_i^j\right)^T y_i^k\right] - \sum_{i=1}^{L}\sum_{p=1}^{L}\sum_{j=1}^{n_i}\sum_{q=1}^{n_p}\left(\frac{1}{n}\left(y_i^j\right)^T y_p^q\right) - \sum_{i=1}^{L}\sum_{j=1}^{n_i}\left(y_i^j\right)^T y_i^j$, it is easy to obtain

$$\mathrm{tr}\left(S_B^{\Phi}\right) - \mathrm{tr}\left(S_W^{\Phi}\right) = \alpha^T G \alpha. \tag{4.52}$$

According to Propositions 1, we can obtain $\mathrm{Dis} = \alpha^T G \alpha$

According to the maximum margin criterion and Proposition 1, we can create an optimization problem constrained by the unit vector α, i.e., $\alpha^T \alpha = 1$, as follows:

$$\max_{\alpha} \quad \alpha^T G \alpha \tag{4.53}$$

$$\text{subject} \quad \text{to} \quad \alpha^T\alpha - 1 = 0 \tag{4.54}$$

In order to solve the above-constrained optimization equation, we apply a Lagrangian

$$L(\alpha, \lambda) = \alpha^T G\alpha - \lambda\left(\alpha^T\alpha - 1\right) \tag{4.55}$$

with the multiplier λ. The derivative of $L(\alpha, \lambda)$ with respect to the primal variables must vanish, that is,

$$\frac{\partial L(\alpha, \lambda)}{\partial \alpha} = (G - \lambda I)\alpha = 0. \tag{4.56}$$

$$\frac{\partial L(\alpha, \lambda)}{\partial \lambda} = 1 - \alpha^T\alpha = 0 \tag{4.57}$$

Hence,

$$G\alpha = \lambda\alpha. \tag{4.58}$$

Proposition 4.2 *Assume α^* is one eigenvector of G corresponding to the largest eigenvalue λ^*, then Dis has a maximum value at α^*.*

Proof For Eq. (4.8), multiply α^T at left side of equation, then

$$\alpha^T G\alpha = \alpha^T\lambda\alpha = \lambda\alpha^T\alpha = \lambda \tag{4.59}$$

Hence,

$$Dis = \lambda \tag{4.60}$$

From this formulation, it is easy to see that *Dis* reaches the largest value when λ reaches the largest value. So *Dis* has a maximum value at α^*, which is an eigenvector of G corresponding to the largest eigenvalue λ^*.

According to Proposition 2, the problem of solving the constrained optimization function is transformed to the problem of solving eigenvalue equation shown in (4.19). The fusion coefficients are equal to the elements of eigenvector of G corresponding to the largest eigenvalue, while G is a matrix which can be calculated by multiple features. As above discussion, discriminant feature fusion finds a discriminating fused feature space, in which data have largest class discriminant. And then, the fusion coefficients are equal to the elements of eigenvector of an eigenvalue problem corresponding to the largest eigenvalue, and the solution of the eigenvalue problem is unique, so the fusion coefficients are unique.

4.6 Three-Dimensional Parameter Selection PCA-Based Face Recognition

4.6.1 Motivation

Subspace-based face recognition is the most successful method of the face recognition approaches. Among the subspace-based face recognition methods, eigenfaces approach based on PCA subspace and fisherfaces approach based on LDA subspace are widely used. Recently, a novel approach named Laplacianfaces based on locality preserving projections (LPP) has been proposed for face recognition [3]. PCA aims to preserve the global structure, and LPP preserves local structure information, and LDA preserves the clustering structure. In order to take full advantage of all of the structure information, we construct a 3D parameter space using the three subspace dimensions as axes. In the 3D parameter space, all above three methods select the principal feature components by searching in the lines or plane only, in other words, they only select the principal feature components in local subspace regions. In our new algorithm, we can search the optimal parameters through the 3D parameter space for selecting the optimal principal components, so it is reasonable to enhance the recognition performance using searching over 3D parameter space instead of only in lines or planes as three standard subspace methods.

In this section, firstly, we analyze the contributions of the three subspace methods, i.e., PCA, LDA, and LPP, and the detailed description of PCA, LDA, and LPP algorithm can be found in [3–5]. PCA aims to preserve the global structure, LPP seeks to preserve the local structure, and LDA aims to find the optimal transformation to preserve the clustering structure. We hope to take advantage of the three structures for the face recognition. Each subspace has the different contribution to feature extraction. Since each subspace has the different contribution to feature extraction, we hope to take full advantage of the three structures provided by PCA, LPP, and LDA for face recognition, and we construct a 3D parameter space using the three subspace dimensions as axes, as shown in Fig. 4.1. Since the selection of the principal components, i.e., the selection of three

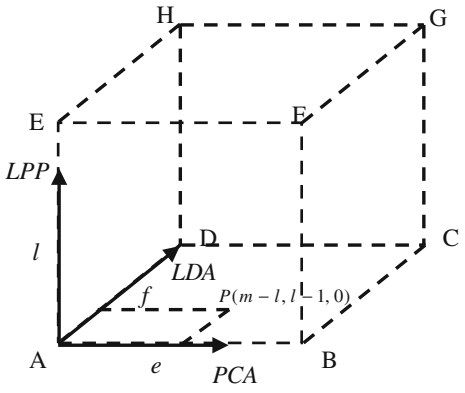


Fig. 4.1 Three-dimensional parameter space for PCA

parameters, greatly influences the recognition performance, we can acquire the different recognition performance by adjusting the three parameters (i.e., the dimensionality of the three subspaces) in the 3D parameters space. The standard PCA, LPP, and LDA only occupy some local lines or areas in the 3D parameter space. Especially, PCA changes the parameters in the e direction on the AB line. Fisherfaces [1] only corresponds to point P $(e = m - l, f = l - 1)$ in Fig. 4.1. Laplacianfaces [3] corresponds to $l - e$ plane in the graph. All these methods only search in the lines or plane, that is, they only search in local regions of 3D parameter space. In our new algorithm, we can search over the 3D parameter space for selecting the optimal principal feature components for face representation and recognition, which enhances the recognition performance. Based on the 3D parameter space, we construct a unified framework to learn in the optimal complex subspaces by searching over the 3D parameter space for face representation and recognition. The algorithm procedure is described as follows:

4.6.2 Method

Given a set of n training samples $\{x_1, \ldots, x_n\}$, let L be the number of classes, and $n_i (i = 1, \ldots, L)$ denote the number of samples in the ith class.

Step 1. Transform the original face data to PCA subspace and select the principal feature components by adjusting the PCA dimension (e) to adjust the contribution of PCA subspace. We can obtain the feature vector $Y_{\text{PCA}} = (y_1, y_2, \ldots y_e)^T$, where $Y_{\text{PCA}} = W_{\text{PCA}} X$.

Step 2. Transform the original face data to LPP subspace and select the principal feature components by adjusting the LPP dimension (l) to adjust the contribution of LPP subspace. LPP feature vector $Y_{\text{LPP}} = (z_1, z_2, \ldots z_l)^T$ can be obtained under the projection matrix $W_{\text{LPP}} = [w_1, w_2, \ldots, w_l]$.

Step 3. Create the complex feature vector using the principal components of PCA and LPP feature subspaces $Y_{\text{PCA}} = (y_1, y_2, \ldots y_e)^T$ and $Y_{\text{LPP}} = (z_1, z_2, \ldots z_l)^T$ as follows:

$$Y_{\text{PCA–LPP}} = \begin{bmatrix} Y_{\text{PCA}} \\ Y_{\text{LPP}} \end{bmatrix} \tag{4.61}$$

Then, apply LDA to the complex PCA and LPP principal feature vector P and select the principal feature components by adjusting the LDA dimension (f) to adjust the contribution of LDA subspace. The principal component of LDA subspace $Y_{\text{PCA–LPP–LDA}} = \left(q_1, q_2, \ldots q_f\right)^T$ can be obtained.

Step 4. The f-dimensional feature vector Y of one sample X under e, l, and f parameters can be acquired by

$$Y = Y_{\text{PCA–LPP–LDA}} \tag{4.62}$$

4.7 Experiments and Discussion

4.7.1 Performance on KPCA and Improved KPCA on UCI Dataset

Firstly, we use the six UCI datasets popular widely in pattern recognition area to testify the performance of the proposed algorithm compared with the KPCA algorithm using the part of training samples and the whole size of samples. In the experiments, we randomly select the one hundred of training samples on each training sample set, especially 20 parts on image and splice dataset. In the experiments, we choose the Gaussian kernel with its parameters determined by the training samples. The experimental results are shown in Tables 4.1, 4.2 and Table 4.3, and the second column shows the error rate of each algorithm on the corresponding dataset. The third column shows the number of training samples in Table 4.1, and the number of training samples in the proposed algorithm in Table 4.2. And in the brackets denote the ratio between the number of training samples of KPCA with the common training method and the proposed training samples. The results show that the proposed algorithm achieves the similar recognition performance, but the proposed algorithm only use the less size of training set. For example, only 8 % training samples are used but only error rate 2.8 % higher than the common methods. Since only small size of training samples is applied in the proposed algorithm, so it will save some place for storing and increase the computation efficiency for KPCA. As the experimental results in Tables 4.2 and 4.3 compared with Table 4.1, KPCA, Sparse KPCA (SKPCA), and sparse data-dependent kernel principal component analysis (SDKPCA), have the similar recognition accuracy but with different number of training samples. SKPCA saves much space of storing training samples on computing the kernel matrix with the lower time-consuming, but achieves the similar recognition accuracy compared with KPCA. The comparison of the results in Tables 4.2 and 4.3 show that SDKPCA achieves the higher recognition accuracy than SKPCA owing to its kernel optimization combined with SKPCA. SDKPCA is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition. This set of experiments show that SDKPCA performs better than SKPCA with the same number of training samples, while SKPCA achieves the

Table 4.1 Recognition performance of KPCA

Datasets	Error rate (%)	Training samples
Banana	13.6 ± 0.1	400
Image	4.8 ± 0.4	1,300
F. Solar	31.4 ± 2.1	666
Splice	8.6 ± 0.8	1,000
Thyroid	2.1 ± 1.0	140
Titanic	22.8 ± 0.3	150

Table 4.2 Recognition performance of SKPCA

Datasets	Error rate (%)	Training samples
Banana	14.2 ± 0.1	120 (30 %)
Image	5.4 ± 0.3	180 (14 %)
F. Solar	34.2 ± 2.3	50 (8 %)
Splice	9.4 ± 0.9	280 (28 %)
Thyroid	2.2 ± 1.3	30 (21 %)
Titanic	23.2 ± 0.5	30 (20 %)

Table 4.3 Recognition Performance of SDKPCA

Datasets	Error rate (%)	Training samples
Banana	13.9 ± 0.2	120 (30 %)
Image	5.1 ± 0.2	180 (14 %)
F. Solar	32.8 ± 2.1	50 (8 %)
Splice	9.0 ± 0.7	280 (28 %)
Thyroid	2.2 ± 1.3	30 (21 %)
Titanic	24.4 ± 0.4	30 (20 %)

similar recognition accuracy but less number of training samples compared with traditional KPCA. The results testify the feasibility of SDKPCA and SKPCA.

4.7.2 Performance on KPCA and Improved KPCA on ORL Database

To quantitatively assess and fairly compare the methods, we evaluate the proposed scheme on ORL [19] and Yale [18] databases under the variable illumination conditions according to a standard testing procedure. ORL face database, developed at the Olivetti Research Laboratory, Cambridge, U.K., is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time, and facial expression. To reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels, and some examples are shown in Fig. 4.1. The experimental results are shown in Table 4.4, SDKPCA performs better than SKPCA under the same number of training samples.

Table 4.4 Performance comparison on ORL face database

Algorithms	Error rate (%)	Training samples
KPCA	15.3 ± 0.8	200
SKPCA	18.4 ± 0.9	120 (60 %)
SDKPCA	17.5 ± 0.7	120 (60 %)

4.7.3 Performance on KPCA and Improved KPCA on Yale Database

Also, we evaluate the proposed scheme on Yale [18] databases under the variable illumination conditions according to a standard testing procedure to quantitatively assess and fairly compare the methods. The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting conditions (left-light, center-light, and right-light), and different facial expressions (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. Similarly, the images from Yale databases are cropped to the size of pixels.

We randomly choose one face image per person as the training sample, and the rest face images are to test the performance of proposed scheme. That is, the rest 9 test samples are to test on ORL face database, while 10 test samples per person are to test the performance on Yale face database. The average recognition accuracy rate is to evaluate the performance of the recognition accuracy, and we implement the experiments for 10 times and 11 times for ORL and Yale face database respectively. As shown in Table 4.5 and Table 4.6 the experimental results show that SDKPCA performs better than SKPCA under the same number of training samples.

In our experiments, we implement our algorithm in the two face databases, ORL face database [9] and Yale face database [8]. The ORL face database, developed at the Olivetti Research Laboratory, Cambridge, U.K., is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time and facial expression. The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains

Table 4.5 Performance comparison on Yale face database

Algorithms	Error rate (%)	Training samples
KPCA	17.8 ± 0.7	75
SKPCA	20.4 ± 0.8	45 (60 %)
SDKPCA	18.7 ± 0.6	45 (60 %)

Table 4.6 Performance comparison on KPCA and RKPCA

Algorithms	Error rate (%)	Training samples
KPCA	3.8 ± 0.4	300
SKPCA	5.4 ± 0.3	110 (37 %)
SDKPCA	4.9 ± 0.4	110 (37 %)

165 grayscale images of 15 individuals. These images are taken under different lighting condition (left-light, center-light, and right-light), and different facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

4.7.4 Performance on Discriminant Parallel KPCA-Based Feature Fusion

In our experiments, to reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels. We randomly select 5 images from each subject, 200 images in total for training, and the rest 200 images are used to test the performance. Similarly, the images from Yale databases are cropped to the size of 100×100 pixels. Randomly selected 5 images per person are selected as the training samples, while the rest 5 images per person are used to test the performance of the algorithms.

From the theory derivation of discriminant fusion in Sect. 4.2, it is easy to predict that the proposed algorithm gives the better performance compared with the classical fusion [4], and here only a set of experiments are implemented for evaluation. Firstly we extract the linear and nonlinear features with PCA and KPCA, and then classify the fused feature of the two features with Fisher classifier. Suppose y_{ij}^1 and y_{ij}^2 $(i = 1, 2, \ldots, C;\ j = 1, 2, \ldots, n_i)$ denote the linear and nonlinear features derived from PCA and KPCA, respectively, the fused feature, $z_i^j = \begin{pmatrix} y_{ij}^1 \\ y_{ij}^2 \end{pmatrix}$ based on the classical fusion [4], while $z_i^j = \sum_{m=1}^{2} a_m y_{ij}^m$ based on discriminant fusion strategy. Here, polynomial kernel and Gaussian kernel with different coefficients are selected for KPCA, and accuracy rate is applied to evaluate the recognition performance.

As results shown in figures, the proposed method gives a higher performance than the classical fusion method [4] under the same kernel parameters for KPCA.

Since the fusion coefficients of the discriminant fusion strategy are obtained by solving the optimization problem based on maximum margin criterion and data have the largest class discriminant in the fused feature space, it is not surprised that discriminant fusion gives a consistently better performance than classical fusion. But besides the above advantages, the following cases are worthy to be considered:

(1) Since the maximum margin criterion is used to create the constrained optimization problem, the fusion strategy is only adapted to the supervised learning.
(2) The fusion coefficients are obtained by solving one eigenvalue problem, which causes the increasing of time-consuming than classical strategy, so it should evaluate the balance of time-consuming and classification accuracy.

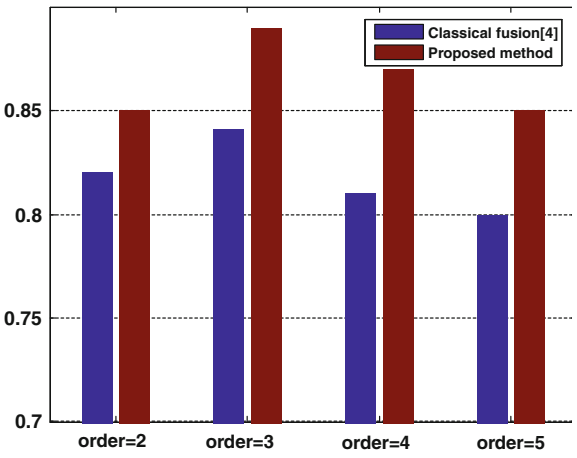


Fig. 4.2 Performance on ORL face database. (polynomial kernel for KPCA)

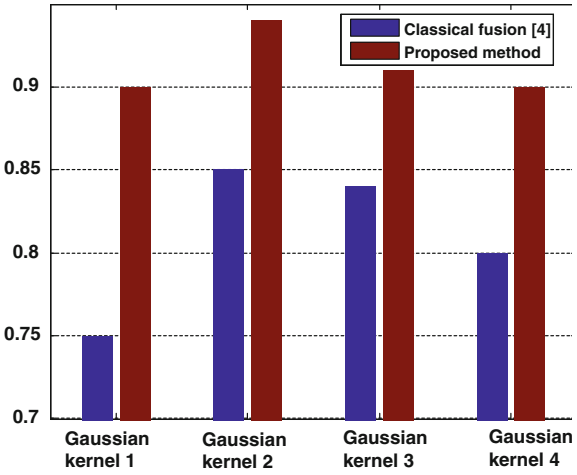


Fig. 4.3 Performance on ORL face database. (Gaussian kernel for KPCA) (Gaussian kernel 1 denotes $\sigma^2 = 1 \times 10^7$; Gaussian kernel 2: $\sigma^2 = 1 \times 10^8$ Gaussian kernel 3: $\sigma^2 = 1 \times 10^9$; Gaussian kernel 4: $\sigma^2 = 1 \times 10^{10}$)

(3) Discriminant fusion strategy is only a linear combination of multiple features with different combination coefficients, so other fusion strategies can be considered to create based on the discriminant analysis (Figs. 4.2, 4.3, 4.4, and 4.5).

4.7.5 Performance on Three-Dimensional Parameter Selection PCA-Based Face Recognition

This section assesses the performance of the proposed method with ORL and Yale face database. Firstly, we implement experiments about selecting principal feature components in 3D parameter space, and secondly, we compare our method with PCA, LPP, and LDA.

Fig. 4.4 Performance on Yale face database. (polynomial kernel for KPCA)

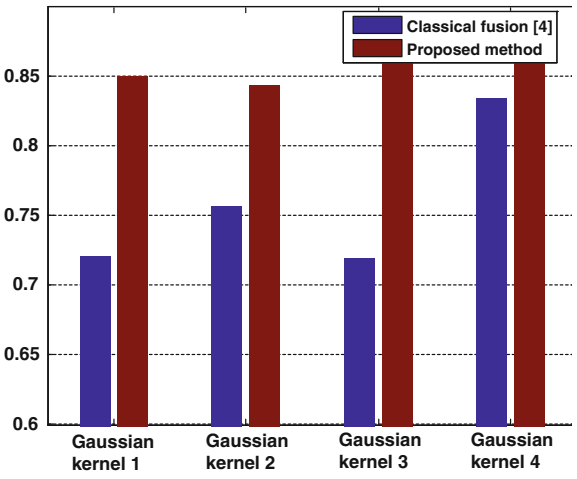


Fig. 4.5 Performance on Yale face database. (Gaussian kernel for KPCA) (Gaussian kernel 3: $\sigma^2 = 1 \times 10^5$; Gaussian kernel 4: $\sigma^2 = 1 \times 10^6$; Gaussian kernel 3: $\sigma^2 = 1 \times 10^7$; Gaussian kernel 4: $\sigma^2 = 1 \times 10^8$)

ORL face database, developed at the Olivetti Research Laboratory, Cambridge, U.K., is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time, and facial expression. The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting condition (left-light, center-light, and right-light), and different facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. To reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels. We randomly select 5 images from each subject, 200 images in total for training, and the rest 200 images are used to test the performance. Similarly, the images from Yale databases are cropped to the size of 100×100 pixels.

In this set of experiments implemented with ORL face database, we select the principal feature components by searching the parameters through the 3D parameter space. We do not search the entire 3D parameter space, but better results can be obtained compared to the standard subspace methods, PCA, LPP, and LDA. The feature vector dimension of the face image is determined by the number of principal feature components of LDA. When the parameters are $e = 20$, $l = 60$, and $f = 7$, respectively, the highest recognition rate is obtained, that is, the dimension of the feature vector is only 7 while the dimension of the original face vector is $48 \times 48 = 2304$. On the other hand, we also select the optimal parameters on the Yale face database. The proposed method obtains the better recognition performance while using the smaller feature vector dimension compared with the standard subspace methods. Since the computational cost for face recognition is proportional to the feature number [4], the proposed method improves the computational efficiency without influencing the recognition performance.

We also test the proposed algorithm with the Yale and ORL face databases compared with the standard subspace approaches, PCA, LDA, and LPP. As shown in Tables 4.7 and 4.8, the proposed algorithm gives the superior recognition performance compared with the standard PCA, LPP, and LDA under the same number of features.

As experimental results shown in Table 4.1, our method gives the highest recognition rate than the other subspace methods, i.e., PCA, LDA, and LPP. We compare the recognition rate under the same feature number. So the proposed method can improve the computational efficiency without influencing the recognition performance. Similarly, as shown in Table 4.2, our method gives the superior results on the ORL face database.

As the experimental results shown, the proposed method gives the superior performance in both enhancing the recognition rate and improving the computational efficiency. On the other hand, we should note that the proposed algorithm does not always perform well at the point of the 3D parameter space where the standard subspace approach performs well. So in practice, we had better search through more points of 3D space as possible to find the optimal point. From the

Table 4.7 Experimental results with Yale face database

Methods	5	10	15	20	25
LDA	0.626	0.626	0.640	0.640	0.640
LPP	0.213	0.440	0.480	0.533	0.693
Our method	0.665	0.820	0.920	0.925	0.940

Table 4.8 Experimental results with ORL face database

Methods	5	10	15	20	25	30
PCA	0.425	0.800	0.800	0.825	0.825	0.825
LDA	0.475	0.610	0.620	0.635	0.640	0.645
LPP	0.185	0.355	0.360	0.440	0.505	0.560
Proposed	0.665	0.850	0.895	0.905	0.925	0.935

above experiments, we can also acquire the following interesting points: (1) All the face recognition approaches mentioned in the experiments perform better in the optimal face subspace than in the original image space. (2) In the 3D parameter space, three standard subspace approaches, PCA, LPP, and LDA, searching only in lines or local regions, while we can take advantage of the three subspaces through searching in the whole parameter space, the proposed algorithm outperforms the standard subspace approaches. So the recognition performance is improved in the new algorithm. (3) The feature number of the face image will influence the computational cost for face recognition, and the proposed method using the small feature number outperforms the standard subspace methods. The proposed method improves the computational efficiency without influencing the recognition performance. On the other hand, in practice, the proposed method can also save the memory consumption.

We present a novel face recognition method by selecting the principal feature components in the 3D parameter space. Firstly, we construct a 3D parameter space using the dimensions of three subspaces (i.e., PCA subspace, LPP subspace, and LDA subspace) as axes. In the 3D parameter space, we can take advantage of the three subspaces through searching in the whole 3D parameter space instead of searching only lines or local region as the standard subspace methods. Then based on the 3D parameter space, we construct a framework for PCA, LPP, and LDA, and the superiority of the proposed algorithm in recognition performance is tested with the ORL and Yale face databases. We expect that the proposed algorithm will provide excellent performance in other areas, such as content-based image indexing and retrieval as well as video and audio classification.

References

1. Chellappa R, Wilson CL, Sirohey S (1995) Human and machine recognition of faces: a survey. Proc IEEE 83:705–739
2. Zhao Z, Huang DS, Sun B-Y (2004) Human face recognition based on multiple features using neural networks committee. Pattern Recognition Lett 25:1351–1358
3. Zhang J, Yan Y, Lades M (1997) Face recognition: eigenface, elastic matching, and neural nets. Proc IEEE 85:1423–1435
4. Brunelli R, Poggio T (1993) Face recognition: features versus templates. IEEE Trans PAMI 15:1042–1052
5. Guo GD, Li SZ, Chan K (2000) Face recognition using support vector machines. In: Proceedings of the IEEE international conference on automatic face and gesture recognition, 2000, pp. 196–201
6. Lee DD, Seung HS (1999) Learning the parts of objects with nonnegative matrix factorization. Nature 401:788–791
7. Li J, Zhou SH, Shekhar C (2003) A comparison of subspace analysis for face recognition. In: Proceedings of the 2003 IEEE international conference on acoustics, speech and signal processing, Hongkong, 2003, pp. 121–124
8. Zhang J, Yan Y, Lades M (1997) Face recognition: eigenface, elastic matching, and neural nets. Proc IEEE 85:1423–1435

9. Sun Y, Wei Z, Wu M, Xiao L, Fei X (2011) Image poisson denoising using sparse representations. Chinese J Electron 39(2):285–290
10. Liu Q, Wang J, Chen W, Qin Z (2011) An automatic feature selection algorithm for high dimensional data based on the stochastic complexity regularization. Chinese J Electron 39(2):370–374
11. Jiang F, Du J, Ge Y, Sui Y, Cao C (2011) Sequence outlier detection based on rough set theory. Chin J Electron 39(2):345–350
12. Hu WC, Yang CY, Huang DY, Huang CH (2011) Feature-based face detection against skin-color like backgrounds with varying illumination. J Inf Hiding Multimedia Signal Process 2(2):123–132
13. Krinidis S, Pitas I (2010) Statistical analysis of human facial expressions. J Inf Hiding Multimedia Signal Process 1(3):241–260
14. Parviz M, Shahram M (2011) Boosting approach for score level fusion in multimodal biometrics based on AUC maximization. J Inf Hiding Multimedia Signal Process 2(1):51–59
15. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
16. Gestel TV, Baesens B, Martens D (2010) From linear to non-linear kernel based classifiers for bankruptcy prediction. Neurocomputing 73(16–18):2955–2970
17. Zhua Q (2010) Reformative nonlinear feature extraction using kernel MSE. Neurocomputing 73(16–18):3334–3337
18. Yang Q, Ding X (2002) Symmetrical PCA in face recognition. In: IEEE ICIP 2002 Proceedings, New York, 2002, pp. 97–100
19. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720

Chapter 5
Kernel Discriminant Analysis Based Face Recognition

5.1 Introduction

Linear discriminant analysis (LDA) is a traditional dimensionality reduction technique for feature extraction. It has been widely used and proven successful in a lot of real-world applications. LDA works well in some cases, but it fails to capture a nonlinear relationship with a linear mapping. In order to overcome this weakness of LDA, the kernel trick is used to represent the complicated nonlinear relationships of input data to develop kernel discriminant analysis (KDA) algorithm. Kernel-based nonlinear feature extraction techniques have attracted much attention in the areas of pattern recognition and machine learning [1–3]. Some algorithms using the kernel trick are developed in recent years, such as kernel principal component analysis (KPCA) [3], KDA [4] and support vector machine (SVM) [5]. KPCA was originally developed by Scholkopf et al. in 1998 [4], while KDA was firstly proposed by Mika et al. in 1999 [6]. KDA has been applied in many real-world applications owing to its excellent performance on feature extraction. Researchers have developed a series of KDA algorithms [7–28]. Because the geometrical structure of the data in the kernel mapping space, which is totally determined by the kernel function, has significant impact on the performance of these KDA methods. The separability of the data in the feature space could be even worse if an inappropriate kernel is used. In order to improve the performance of KDA, many methods of optimizing the kernel parameters of the kernel function are developed in recent years [16–18]. However, choosing the parameters for kernel just from a set of discrete values of the parameters doesn't change the geometrical structures of the data in the kernel mapping space. In order to overcome the limitation of the conventional KDA, we introduce a novel kernel named quasiconformal kernel which were widely studied in the previous work [29–32], where the geometrical structure of data in the feature space is changeable with the different parameters of the quasiconformal kernel. The optimal parameters are computed through optimizing an objective function designed with the criterion of maximizing the class discrimination of the data in the kernel mapping space. Consequently AQKDA is more adaptive to the input data for classification

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_5,

than KDA. Experiments implemented on ORL and YALE face databases demonstrate the feasibility of the proposed algorithm compared with conventional KDA algorithm.

5.2 Kernel Discriminant Analysis

KDA is based on a conceptual transformation from the input space into a nonlinear high-dimensional feature space. Supposed that M training samples $\{x_1, x_2, \ldots, x_M\}$ with L class labels take values in an N-dimensional space $\mathbb{R}^N$, the data in $\mathbb{R}^N$ are mapped into a feature space F via the following nonlinear mapping:

$$\Phi : \mathbb{R}^N \to F,\ x \mapsto \Phi(x) \tag{5.1}$$

Consequently in the feature space F Fisher criterion (FC) is defined by

$$J(V) = \frac{V^T S_B^\Phi V}{V^T S_T^\Phi V} \tag{5.2}$$

where V is the discriminant vector, and S_B^Φ and S_T^Φ are the between classes scatter matrix and the total population scatter matrix respectively [13]. According to the Mercer kernel function theory, any solution V belongs to the span of all training pattern in $\mathbb{R}^N$. Hence there exist coefficients $c_p\ (p = 1, 2, \ldots, M)$ such that

$$V = \sum_{p=1}^{M} c_p \Phi(x_p) = \Psi\alpha \tag{5.3}$$

where $\Psi = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_M)]$ and $\alpha = [c_1, c_2, \ldots, c_M]^T$. Suppose the data are centered, FC is transformed into

$$J(\alpha) = \frac{\alpha^T KGK\alpha}{\alpha^T KK\alpha} \tag{5.4}$$

where $G = \mathrm{diag}(G_1, G_2, \ldots, G_L)$ and G_i is an $n_i \times n_i$ matrix whose elements are $\frac{1}{n_i}$, and K is the kernel matrix calculated by a basic kernel $k(x, y)$. The criterion given in (5.4) attains its maximum for the orthonormal vectors. There are numerous algorithms to find this optimal subspace and an orthonormal basis for it.

5.3 Adaptive Quasiconformal Kernel Discriminant Analysis

Choosing the appreciate kernels is a crucial problem for KDA. AQKDA uses the quasiconformal kernel to improve KDA. The quasiconformal kernel was used to improve SVM and other kernel-based learning algorithm in the

previous works [31, 32]. KDA is implemented in the high-dimensional quasiconformal kernel space to develop AQKDA algorithm. In AQKDA, FC in the quasiconformal kernel space is defined by

$$J(\alpha) = \frac{\alpha^T \widehat{K} G \widehat{K} \alpha}{\alpha^T \widehat{K} \widehat{K} \alpha} \tag{5.5}$$

where $\widehat{K}$ is the matrix calculated with the quasiconformal kernel. The definition of the quasiconformal kernel $\widehat{k}(x, y)$ is described as follows.

$$\widehat{k}(x, y) = f(x) f(y) k(x, y) \tag{5.6}$$

where $f(x)$ is a positive real valued function x, and $k(x, y)$ is a basic kernel, e.g., polynomial kernel and Gaussian kernel. Amari and Wu [32] expanded the spatial resolution in the margin of a SVM by using $f(x) = \sum_{i \in SV} a_i e^{-\delta \left\| x - \widetilde{x_i} \right\|^2}$, where $\widetilde{x_i}$ is the ith support vector, and SV is a set of support vector, and a_i is a positive number representing the contribution of $\widetilde{x_i}$, and δ is a free parameter. We generalize Amari and Wu's method as

$$f(x) = b_0 + \sum_{n=1}^{N_{XV}} b_n e(x, \widetilde{x}_n) \tag{5.7}$$

where $e(x, \widetilde{x}_n) = e^{-\delta \left\| x - \widetilde{x_n} \right\|^2}$, and δ is a free parameter, and $\widetilde{x}_n$ are called the "expansion vectors (XVs)", and N_{XV} is the number of XVs, and b_n $(n = 0, 1, 2, \ldots, N_{XV_s})$ are the "expansion coefficients" associated with $\widetilde{x}_n$ $(n = 0, 1, 2, \ldots, N_{XV_s})$. The definition of the quasiconformal kernel shows that the geometrical structure of the data in the kernel mapping space is determined by the expansion coefficients with the determinative XVs and free parameter. Moreover the criterion of AQKDA in (5.5) means two stages of AQKDA, i.e., optimizing the quasiconformal kernel and finding the projection matrix. The key issue of optimizing the quasiconformal kernel is to design an objective function. As shown in (5.6) and (5.7), given the XVs, the free parameter and the basic kernel, the quasiconformal kernel is the function with the variable "expansion coefficients". So the first thing is to choose all these procedural parameters. As shown in Table 5.1, four methods of choosing XVs are proposed to solve the expansion coefficients. XVs 1 is similar to the method in the previous work [30] only considers the nearest neighbor criterion in the original high dimensional space without considering the label information. KDA is a supervised learning problem, and the label information of each training sample can be considered for feature extraction. XVs 2 considers the class label information of all training samples. But XVs 1 and XVs 2 endure store space and computation efficiency problem when the size of training set is very large. In order to solve this problem, we proposed the method XVs 3 and XVs 4. XVs 3 and XVs 4 only consider the distribution of distance between each sample and the mean of the sample belongs to the same class. Especially, the goal of XVs 3 lies in that the samples from the same class

Table 5.1 Four methods of choosing expansion vectors (XVs)

XVs 1	$e(x,\tilde{x}_n) = e(x,x_n) = \begin{cases} 1 \\ e^{-\delta\|x-x_n\|^2} \end{cases}$	x and x_n with the same class label information x and x_n with the different class label information
XVs 2	$e(x,\tilde{x}_n) = e(x,x_n) = e^{-\delta\|x-x_n\|^2}$	
XVs 3	$e(x,\tilde{x}_n) = e(x,x_n) = \begin{cases} 1 \\ e^{-\delta\|x-\overline{x_n}\|^2} \end{cases}$	The class label information of x and $\overline{x_n}$ is same; The class label information of x and $\overline{x_n}$ is different;
XVs 4	$e(x,\tilde{x}_n) = e(x,\overline{x_n}) = e^{-\delta\|x-\overline{x_n}\|^2}$	

Notes XVs 1 All training samples as XVs with considering the label information of each sample; XVs 2 All training samples as XVs without considering the label information of each sample; XVs 3 Mean of each class as XVs with considering the label information of each sample; XVs 4 Mean of each class as XVs without considering the label information of each sample. $\overline{x_n}$ is the mean of each class

centralize in high dimension space after the quasiconformal mappings. So the distance between any two samples from the same class is zero, i.e., $e(x,x_n) = 1$. Moreover, the free parameter and the basic kernel are selected through the experiments with the cross-validation method.

Now we design an objective function to find the adaptive expansion coefficients varied with the input data for the quasiconformal kernel. Given the free parameter δ and the expansion vectors $\{\widetilde{x}_i\}_{i=1,2,\ldots,N_{XVs}}$, we create the matrix

$$E = \begin{bmatrix} 1 & e(x_1,\widetilde{x}_1) & \cdots & e(x_1,\widetilde{x}_{N_{XVs}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e(x_M,\widetilde{x}_1) & \cdots & e(x_M,\widetilde{x}_{N_{XVs}}) \end{bmatrix} \tag{5.8}$$

Let $\beta = [b_0, b_1, b_2, \ldots, b_{N_{\mathrm{XVs}}}]^T$ and $\Lambda = \mathrm{diag}(f(x_1), f(x_2), \ldots, f(x_M))$ $i = 0, 1, 2, \ldots, N_{\mathrm{XVs}}$, we obtain

$$\Lambda 1_M = E\beta \tag{5.9}$$

where 1_M is a M-dimensional vector whose entries equal to unity.

Proposition 1 *Let K and $\widehat{K}$ denote the basic kernel matrix and quasiconformal kernel matrix respectively, then $\widehat{K} = \Lambda K \Lambda$.* □

We expect to solve the expansion coefficient vector β through optimizing an objective function designed for measuring the class separability of the data in the feature space. We use FC and Maximum Margin Criterion (MMC) to measure the class separability for the objective function. FC measures the class separability in the high-dimensional feature space, and FC is widely used to feature extraction in the previous works [11, 13]. The FC is defined as

$$J_{\mathrm{Fisher}} = \frac{tr(S_B^{\Phi})}{tr(S_W^{\Phi})} \tag{5.10}$$

where S_B^{Φ} and S_W^{Φ} denote the between-class scatter matrix and the within-class scatter matrix in the high-dimensional space respectively, and tr and J_{Fisher} denote the trace of a matrix and the Fisher scalar of measuring the class separability of the data.

Proposition 2 *Assume* $Kc_{ij} \quad (i, j = 1, 2, \ldots, L)$ *is a kernel matrix calculated with the ith and jth class samples and the kernel matrix* K_{total} *with its elements* k_{ij} *calculated with pth and qth samples. Then* $tr\left(S_B^{\Phi}\right) = 1_n^T B 1_n$ *and* $tr\left(S_W^{\Phi}\right) = 1_n^T W 1_n$ *where* $B = \mathrm{diag}\left(\frac{1}{n_1}Kc_{11}, \frac{1}{n_2}Kc_{22}, \ldots, \frac{1}{n_L}Kc_{LL}\right) - \frac{1}{n}K_{\mathrm{total}}$ *and* $W = \mathrm{diag}(k_{11}, k_{22}, \ldots, k_{nn}) - \mathrm{diag}\left(\frac{1}{n_1}Kc_{11}, \frac{1}{n_2}Kc_{22}, \ldots, \frac{1}{n_L}Kc_{LL}\right)$. □

Consequently, FC in (5.10) is rewritten as

$$J_{\mathrm{Fisher}} = \frac{1_n^T B 1_n}{1_n^T W 1_n} \tag{5.11}$$

Then the traces of the between-class scatter matrix and the within-class scatter matrix in the quasiconformal kernel space respectively are $tr\left(\widetilde{S}_B^{\Phi}\right) = 1_n^T \widetilde{B} 1_n$ and $tr\left(\widetilde{S}_W^{\Phi}\right) = 1_n^T \widetilde{W} 1_n$. Where $\widetilde{B} = \Lambda B \Lambda$ and $\widetilde{W} = \Lambda W \Lambda$. FC $\widetilde{J}_{\mathrm{Fisher}}$ in the quasiconformal kernel mapping space is defined as follows.

$$\widetilde{J}_{\mathrm{Fisher}} = \frac{1_n^T \Lambda B \Lambda 1_n}{1_n^T \Lambda W \Lambda 1_n} \tag{5.12}$$

Consider the Eq. (5.12) together with the Eq. (5.9), we obtain

$$\widetilde{J}_{\mathrm{Fisher}} = \frac{\beta^T E^T B E \beta}{\beta^T E^T W E \beta} \tag{5.13}$$

$E^T B E$ and $E^T W E$ are the constant matrices as soon as the basic kernel, XVs and the free parameter are determined. Say, $\widetilde{J}_{\mathrm{Fisher}}$ is a function with its variable β. Thus we create an objective function constrained by the unit vector β, i.e., $\beta^T \beta = 1$, to maximize $\widetilde{J}_{\mathrm{Fisher}}$, which can be described as

$$\begin{aligned} \max \quad & \frac{\beta^T E^T B E \beta}{\beta^T E^T W E \beta} \\ \text{subject to} \quad & \beta^T \beta - 1 = 0 \end{aligned} \tag{5.14}$$

Based on the FC the optimal solution must be obtained by iteration updating algorithm owing to the singular problem of matrix, and detailed algorithm can be found in [30]. The limitation of this method lies in the difficult in finding the unique optimal solution and the high time consuming owing iteration updating algorithm. So we consider the other criterion of seeking the optimal β, i.e., MMC, to solve the limitation problem.

MMC [30] is used to extract the feature by maximizing the average margin between the different classes of data in the high-dimensional feature space. The average margin between two classes c_i and c_j in the feature space is defined as

$$\text{Dis} = \frac{1}{2n}\sum_{i=1}^{L}\sum_{j=1}^{L} n_i n_j d\left(c_i, c_j\right) \tag{5.15}$$

where $d\left(c_i, c_j\right) = d\left(m_i^{\Phi}, m_j^{\Phi}\right) - S(c_i) - S\left(c_j\right),\ i, j = 1, 2, \ldots, L$, denotes the margin between any two classes, and $S(c_i),\ i = 1, 2, \ldots, L$, denotes the measure of the scatter of the class, and $d\left(m_i^{\Phi}, m_j^{\Phi}\right),\ i, j = 1, 2, \ldots, L$, denotes the distance between the means of two classes. Supposed that $tr\left(S_i^{\Phi}\right)$ measures the scatter of the class i, then $tr\left(S_i^{\Phi}\right) = \frac{1}{n_i}\sum_{p=1}^{n_i}\left(\Phi(x_i^p) - m_i^{\Phi}\right)^T\left(\Phi(x_i^p) - m_i^{\Phi}\right)$, where $S_i^{\Phi}, \quad i = 1, 2, \ldots, L$, denotes the within-class scatter matrix of class i. Say, $S(c_i) = tr\left(S_i^{\Phi}\right). \quad i = 1, 2, \ldots, L$.

Proposition 3 *Let* $tr\left(S_B^{\Phi}\right)$ *and* $tr\left(S_W^{\Phi}\right)$ *denote the trace of between-class scatter matrix and the within-class scatter matrix respectively, then* $Dis = tr\left(S_B^{\Phi}\right) - tr\left(S_W^{\Phi}\right)$. □

Considering Proposition 2 together with Proposition 3, we obtain

$$Dis = 1_n^T(B - W)1_n \tag{5.16}$$

Then the average margin $\tilde{Dis}$ in the quasiconformal kernel mapping space is written as

$$\widetilde{Dis} = \beta^T E^T(B - W)E\beta \tag{5.17}$$

After a basic kernel and XVs and the free parameter are determined, $E^T(B - W)E$ is a constant matrix. $\widetilde{Dis}$ is a function with its variable β. An optimization function of maximizing $\widetilde{Dis}$ constrained by the unit vector β, i.e., $\beta^T\beta = 1$ is defined as

$$\begin{array}{ll} \max & \beta^T E^T(B - W)ME\beta \\ \text{subject to} & \beta^T\beta - 1 = 0 \end{array} \tag{5.18}$$

The solution of the above constrained optimization problem is found with the so-called Lagrangian method, and then it is transformed into the following eigenvalue equation with the parameter λ.

$$E^T(B - W)E\beta = \lambda\beta \tag{5.19}$$

Solving the optimal expansion coefficient vector β^* is equal to the eigenvector of $E^T(B - W)E$ corresponding to the largest eigenvalue. Now in order to find the optimal β^*, we use an effective way similar to the previous work [36] to calculate the eigenvector of $E^T(B - W)E$ with avoiding the small size sample (SSS) problem.

The objective function designed based on FC and MMC is to maximize the class separability of the data in the optimal quasiconformal kernel mapping space.

After solving the adaptive parameters for the quasiconformal kernel, we implement KDA algorithm in the optimal quasiconformal kernel mapping with the FC with feature extraction shown in the Eq. (5.5). The optimal projection α from the feature space to the projection subspace is calculated by the same way as KDA. The procedure of AQKDA is shown as follows.

Input: A set of N-dimensional training vectors $\{x_1, x_2, \ldots, x_M\}$.
Output: A low d-dimensional representation y of x.

Step 1. Calculate E, B, W;
Step 2. Solve β^* using the Eq. (5.14) (if using FC) or the Eq. (5.19) (if using MMC);
Step 3. Calculate $\widehat{K}$ with β^*;
Step 4. Calculate the projection matrix $A = [\alpha_1, \alpha_2, \ldots, \alpha_d]$ by solving the Eq. (5.5);
Step 5. Extract the feature vector $y = A^T \left[\widetilde{k}(x_1, x), \widetilde{k}(x_2, x), \ldots, \widetilde{k}(x_M, x)\right]^T$.

5.4 Common Kernel Discriminant Analysis

Current feature extraction methods can be classified to signal processing and statistical learning methods. On signal processing based methods, feature extraction based Gabor wavelets are widely used to represent the face image [3, 4], because the kernels of Gabor wavelets are similar to two-dimensional receptive field profiles of the mammalian cortical simple cells, which captures the properties of spatial localization, orientation selectivity, and spatial frequency selectivity to cope with the variations in illumination and facial expressions. In this chapter, we proposed a novel face recognition method called *common Gabor vector* (CGV) method to solve the poses, illuminations, expressions problems in the practical face recognition system. Firstly, we analyze the limitations of the traditional discriminative common vector (DCV) [27] on the nonlinear feature extraction for face recognition owing to the variations in poses, illuminations and expressions. In order to overcome this limitation, we extend DCV with kernel trick with the space isomorphic mapping view in the kernel feature space and develop a two-phase algorithm, KPCA + DCV, for face recognition. And then, in order to enhance the recognition performance, we apply Gabor wavelet to extract the Gabor features of face images firstly, and then extract the CGV with the proposed kernel discriminant common vector analysis (KDCV). Since CGV method extracts the nonlinear features with Gabor wavelets and KDCV and the influences of the variations in illuminations, poses and expressions are decreased, the high recognition accuracy is achieved which is tested on ORL and Yale face databases.

5.4.1 Kernel Discriminant Common Vector Analysis with Space Isomorphic Mapping

In this section, firstly we analyze the traditional DCV algorithm from a novel view, i.e., isomorphic mapping and then extend it with kernel trick. Secondly, we apply it to face recognition to present the framework of CGV.

The main idea of DCV is to develop the common properties of all classes through eliminating the differences of the samples within the class. The within-class scatter matrix is created for the common vectors instead of using a given class's own scatter matrix. DCV obtains the common vectors with the subspace methods and the Gram-Schmidt orthogonalization procedure to propose DCVs. We present the novel formulation of common vector analysis from space isomorphic mapping view as follows. The Fisher discriminant analysis in Hilbert space H based on FC function is defined as

$$J(\varphi) = \frac{\varphi^T S_B \varphi}{\varphi^T S_W \varphi} \tag{5.20}$$

where $S_W = \sum_{i=1}^{C} \sum_{j=1}^{N} \left(x_i^j - m_i\right)\left(x_i^j - m_i\right)^T$ and $S_B = \sum_{i=1}^{C} N(m_i - m)(m_i - m)^T$ are both positive in the Hilbert space H. Specially, $\varphi^T S_W \varphi = 0$, the FC is transformed as

$$J_b(\varphi) = \varphi^T S_B \varphi, (\|\varphi\| = 1) \tag{5.21}$$

The data become well separable under criterion (2) where $\varphi^T S_W \varphi = 0$, which are analyzed with space isomorphic mapping in the Hilbert space as follows [28]. Supposed a compact and self-adjoint operator on Hilbert space H, then its eigenvector forms an orthonormal basis for H, $\left(H = \Psi \oplus \Psi^{\perp}\right)$. Then an arbitrary vector φ $(\varphi \in H)$ is represented $\varphi = \phi + \zeta$ $\left(\phi \in \Psi,\ \zeta \in \Psi^{\perp}\right)$, and the mapping $P : H \to \Psi$ where $\varphi = \phi + \zeta \to \phi$ and the orthogonal projection ϕ of φ onto H. According to $P : H \to \Psi$ with $\varphi = \phi + \zeta \to \phi$ the criterion in (2) is transformed as

$$J_b(\varphi) = J_b(\phi) \tag{5.22}$$

So, P is a linear operator from H onto its subspace Ψ. Supposed $\alpha_1, \alpha_2, \ldots, \alpha_m$ of S_W, $\Omega_w = \text{span}\{\alpha_1, \alpha_2, \ldots, \alpha_q\}$ and $\overline{\Omega}_w = \text{span}\{\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m\}$ are the range space and null space of S_W respectively, where $\mathbb{R}^m = \overline{\Omega}_w \oplus \Omega_w$, $q = \text{rank}(S_w)$. Since Ω_w and $\overline{\Omega}_w$ are isomorphic to $\mathbb{R}^q$ and $\mathbb{R}^p$ $(p = m - q)$ respectively, $P_1 = \left(\alpha_1, \alpha_2, \ldots, \alpha_q\right)$ and $P_2 = \left(\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m\right)$ the corresponding isomorphic mapping is defined by

$$\varphi = P_2 \theta \tag{5.23}$$

Then criterion in (3) is converted into

$$J_b(\theta) = \theta^T \widehat{S}_b \theta, (\|\theta\| = 1) \tag{5.24}$$

where $\widehat{S}_b = P_2^T S_b P_2$. The stationary points $\mu_1, \ldots, \mu_d (d \leq c-1)$ of $J_b(\theta)$ are the orthonormal eigenvectors of $\widehat{S}_b$ corresponding to the d largest eigenvalues. The optimal irregular discriminant vectors with respect to $J_b(\varphi)$, $\varphi_1, \varphi_2, \ldots, \varphi_d$ are acquired through $\varphi_i = P_2 \mu_i (i = 1, \ldots, d)$. So the irregular discriminant feature vector y of the input vector x is obtained by

$$y = (\varphi_1, \varphi_2, \ldots, \varphi_d)^T x \tag{5.25}$$

Supposed that the stationary points $\mu_1, \ldots, \mu_d (d \leq c-1)$ of $J_b(\theta)$ be the orthonormal eigenvectors of $\widehat{S}_b$ corresponding to the d largest eigenvalues, then

$$\widehat{S}_b \mu_i = \lambda \mu_i \quad i = 1, 2, \ldots, d \tag{5.26}$$

Then

$$P_2 \widehat{S}_b \mu_i = \lambda P_2 \mu_i \quad i = 1, 2, \ldots, d \tag{5.27}$$

Since $P_2 = \left(\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m\right)$, $P_2^T P_2 = c$ where c is a constant value. Then

$$P_2 \widehat{S}_b \left(P_2^T P_2\right) \mu_i = \lambda \left(P_2^T P_2\right) P_2 \mu_i \quad i = 1, 2, \ldots, d \tag{5.28}$$

That is

$$\left(P_2 \widehat{S}_b P_2^T\right) P_2 \mu_i = \lambda \left(P_2^T P_2\right) P_2 \mu_i \quad i = 1, 2, \ldots, d \tag{5.29}$$

Let $w = P_2 \mu_i$ and $\lambda_w = c\lambda$, then

$$\left(P_2 \widehat{S}_b P_2^T\right) w_i = \lambda_w w_i \quad i = 1, 2, \ldots, d \tag{5.30}$$

where w is an eigenvector of $\overline{S_b} = P_2 \widehat{S}_b P_2^T$ corresponding to d largest eigenvalue. Then

$$\overline{S_b} = \sum_{i=1}^{C} N \left(P_2 P_2^T m_i - P_2 P_2^T m\right) \left(P_2 P_2^T m_i - P_2 P_2^T m\right)^T \tag{5.31}$$

In the null space of S_W, i.e., $\overline{\Omega}_w = span\left\{\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m\right\}$, there

$$P_2 P_2^T S_w P_2 P_2^T = \sum_{i=1}^{C} \sum_{j=1}^{N} \left(y_i^j - u_i\right) \left(y_i^j - u_i\right)^T = 0 \tag{5.32}$$

where $P_2 = \left(\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m\right)$ and $P_2^T S_w P_2 = 0$, $u_i = P_2 P_2^T m_i$, $u = P_2 P_2^T m$, so $y_i^j = P_2 P_2^T x_i^j$, and let $Y_C = \left[y_1^1 - u_1 \, y_1^1 - u_1 \ldots y_C^N - u_C\right]$ then $Y_C Y_C^T = 0$. Say that

for any sample y_i^j in ith class, we can obtain the same unique vector u_i for all samples of the same class. The Eq. (5.12) can be transformed into

$$\overline{S_b} = \sum_{i=1}^{C} N(u_i - u)(u_i - u)^T \tag{5.33}$$

Let $x_{\text{com}}^i = P_2 P_2^T x_i^j$ then

$$S_{\text{com}} = \sum_{i=1}^{C} N\left(x_{\text{com}}^i - u_{\text{com}}\right)\left(x_{\text{com}}^i - u_{\text{com}}\right)^T \tag{5.34}$$

where $u_{\text{com}} = \frac{1}{C}\sum_{i=1}^{C} x_{\text{com}}^i$. For a input vector x, the discriminant feature vector y can be obtained as

$$y = (w_1, w_2, \ldots, w_d)^T x \tag{5.35}$$

where $w_1, w_2, \ldots, w_d,\ d \leq C - 1$, are the orthonormal eigenvectors of S_{com}.

5.4.2 Gabor Feature Analysis

Gabor wavelets are optimally localized in the space and frequency domains, and the two-dimensional Gabor function $g(x, y)$ is defined by

$$g(x, y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) \exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi j\omega x\right] \tag{5.36}$$

Its Fourier transform $G(u, v)$ can be written by:

$$G(u, v) = \exp\left\{-\frac{1}{2}\left[\frac{(u-\omega)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2}\right]\right\} \tag{5.37}$$

where ω is the center frequency of $G(u, v)$ along the u axis, $\sigma_u = \frac{\sigma_x}{2\pi}$ and $\sigma_v = \frac{\sigma_y}{2\pi}$. σ_x and σ_y characterize the spatial extent along x and y axes respectively, while σ_u and σ_v characterize the band width along u and v axes respectively. A self-similar filter dictionary is obtained through the proper dilations and rotations of $g(x, y)$ with the following function:

$$g_{mn}(x, y) = a^{-m} g(x', y'),\ a > 1,\ m, n \in \mathrm{Z} \tag{5.38}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = a^{-m} \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5.39}$$

where $\theta = n\pi/K$, and K is the total number of orientations, and a^{-m} is the scale factor, and $g(x, y)$ is the basic Gabor wavelet. Let U_l and U_k denote the lower and

upper center frequencies of interest, K be the number of orientations, and S be the number of scales in the multi-resolution decomposition respectively. The filter parameters can be obtained by

$$a = (U_h/U_l)^{\frac{1}{S-1}} \tag{5.40}$$

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}} \tag{5.41}$$

$$\sigma_v = \tan\left(\frac{\pi}{2k}\right)\left[U_h - (2\ln 2)\frac{\sigma_u^2}{U_h}\right]\left[2\ln 2 - \frac{(2\ln 2)^2\sigma_u^2}{U_h^2}\right]^{-\frac{1}{2}} \tag{5.42}$$

where $\omega = U_h$ and $m = 0, 1, \ldots, S-1$.

For a face image $I(x, y)$, its Gabor features are achieved as

$$W_{mn}(x, y) = \int I(x, y) g_{mn}^*(x - \xi, y - \eta) d_\xi d_\eta \tag{5.43}$$

As the above discussion, the Gabor features are robust to variations in illumination and facial expression changes.

5.4.3 Algorithm Procedure

The CGV method is divided into two steps, i.e., feature extraction and classification. In feature extraction, For the input image I, firstly, the Gabor feature vector F_g extracted with are extracted with Gabor, secondly, the CGV z is calculated with the proposed KDCV. In recognition, Nearest Neighbor Classifier (NNC) is used to classify the CGV from a new face image for recognition.

Step 1. For the input image I, calculate the Gabor feature matrix with Eq. (5.24) and transform it to one vector $Fg(I)$.

Step 2. Compute the orthonormal eigenvector of S_W, $\alpha_1, \alpha_2, \ldots, \alpha_m$, and construct the matrix $P_2 = \left(\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m\right)$, where $q = \text{rank}(S_w)$ and the class scatter matrix S_W is constructed with the KPCA-based vector y of Gabor features from the training set.

Step 3. Create the common between scatter matrix S_{com} with Eq. (5.15) and the common vector $z_{\text{com}} = P_2 P_2^T y$.

Step 4. Compute the orthonormal eigenvector of S_{com}, $w_1, w_2, \ldots, w_d$, $(d \leq C-1)$ and the project matrix $W = [w_1, w_2, \ldots, w_d]$.

Step 5. Compute the CGV $z_{\text{com}} = W^T y$.

5.5 Complete Kernel Fisher Discriminant Analysis

5.5.1 Motivation

Two issues are essential in face recognition algorithms: what features are used to represent a face image, and how to classify a new face image based on this representation. This section details the novel method for face recognition. Firstly, desirable Gabor features are derived from Gabor representations of face images. Secondly, the Complete Kernel Fisher Discriminant (CKFD) analysis with fractional power polynomial (FPP) Models based classifier is performed on the Gabor wavelet representation for classification. Kernel Fisher Discriminant (KFD) is another nonlinear subspace analysis method, which combines the kernel trick with LDA. After the input data are mapped into F with the kernel trick, LDA is performed in F to yield nonlinear discriminant features of the input data. In the CKFD algorithm, two kinds of discriminant information, regular and irregular information, makes the KFD more powerful as a discriminator.

5.5.2 Method

LDA aims to seek the projection matrix to distinguish the face class, through maximizing the between-class scatter S_b but minimizing the within-class S_w of the data in the projection subspace. In LDA, the FC functions $J(\varphi)$ and $J_b(\varphi)$ can be defined by:

$$J(\varphi) = \frac{\varphi^T S_b \varphi}{\varphi^T S_w \varphi}, \quad (\varphi \neq 0) \tag{5.44}$$

$$J_b(\varphi) = \varphi^T S_b \varphi, \quad (\|\varphi\| = 1) \tag{5.45}$$

CKFD performs LDA in the KPCA-transformed space R^m. The strategy of CKFD is to split the space R^m into two subspaces, i.e. the null space and the range space of S_w. Then the FC is used to extract the regular discriminant vectors from the range space, and the between-class scatter criterion is used to extract the irregular discriminant vectors from the null space.

Given the orthonormal eigenvector of S_w, $\alpha_1, \alpha_2, \ldots, \alpha_m$, $\Omega_w = \text{span}\{\alpha_1, \alpha_2, \ldots, \alpha_q\}$ is the range space and $\overline{\Omega_w} = \text{span}\{\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_{q+m}\}$ is the null space of S_w and $R^m = \overline{\Omega_w} \oplus \Omega_w$, where $q = \text{rank}(S_w)$. Since Ω_w and $\overline{\Omega_w}$ are isomorphic to Euclidean space $\mathbb{R}^q$ and Euclidean space $\mathbb{R}^p (p = m - q)$ respectively, and let $P_1 = (\alpha_1, \alpha_2, \ldots, \alpha_q)$ and $P_2 = (\alpha_{q+1}, \alpha_{q+2}, \ldots, \alpha_m)$, we can define the corresponding isomorphic mapping by:

$$\varphi = P_1 \theta \tag{5.46}$$

$$\varphi = P_2\theta \tag{5.47}$$

Under the mapping denoted by (5.11) and (5.12), the FC function $J(\varphi)$ and the between-class scatter criterion $J_b(\varphi)$ are converted into the following equations respectively:

$$J(\theta) = \frac{\theta^T \widetilde{S}_b \theta}{\theta^T \widetilde{S}_w \theta}, \quad (\theta \neq 0) \tag{5.48}$$

$$J_b(\theta) = \theta^T \widehat{S}_b \theta, \quad (\|\theta\| = 1) \tag{5.49}$$

where $\widetilde{S}_b = P_1^T S_b P_1$, $\widetilde{S}_w = P_2^T S_w P_2$ and $\widehat{S}_b = P_2^T S_b P_2$. The stationary points $\upsilon_1, \ldots, \upsilon_d (d \leq c-1)$ of $J(\theta)$ are the orthonormal eigenvectors of $\widetilde{S}_b\theta = \lambda \widetilde{S}_w \theta$ corresponding to the d largest eigenvalues. The optimal regular discriminant vectors $\widetilde{\xi}_1, \widetilde{\xi}_2, \ldots, \widetilde{\xi_d}$ can be acquired by $\widetilde{\xi}_i = P_2\upsilon_i (i = 1, \ldots, d)$ according to (5.12). In the similar way, the stationary points $\mu_1, \ldots, \mu_d (d \leq c-1)$ of $J_b(\theta)$ are the orthonormal eigenvectors of $\widehat{S}_b$ corresponding to the d largest eigenvalues. The optimal irregular discriminant vectors with respect to $J_b(\theta)$, $\widehat{\xi}_1, \widehat{\xi}_2, \ldots, \widehat{\xi}_d$ can be acquired by $\widehat{\xi}_i = P_2\mu_i$ $(i = 1, \ldots, d)$.

For a sample x, after KPCA, we can obtain the sample vector y in the KPCA-transformed space. The regular discriminant feature vector z_1 and the irregular discriminant feature vector z_2 of the sample y are defined respectively as follows:

$$z_1 = \left(\widetilde{\xi}_1, \widetilde{\xi}_2, \ldots, \widetilde{\xi}_d\right)^T y \tag{5.50}$$

$$z_2 = \left(\widehat{\xi}_1, \widehat{\xi}_2, \ldots, \widehat{\xi}_d\right)^T y \tag{5.51}$$

Finally, we fuse the z_1 and z_2 with the proper similarity measure.

The main idea of CKFD is to map the input data into a feature space by a nonlinear mapping where inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly. Three kinds of kernel functions used in the kernel-based learning machine are polynomial kernels, Gaussian kernels, and sigmoid kernels. Other than above three kinds of Kernels, the FPP models, are proposed.

$$k(x, y) = (x \cdot y)^d \quad (0 < d < 1) \tag{5.52}$$

A FPP is not necessarily a kernel, as it might not define a positive semi-definite Gram matrix, thus the FPPs are called models rather kernels.

Popular similarity measures include Euclidean distance measure δ_L, $L1$ distance measure δ_{L_1}, and cosine similarity measure $\delta_{\cos}$, which are defined as follows, respectively:

$$\delta_L = \|x - y\| \tag{5.53}$$

$$\delta_{L_1} = \sum_i |x_i - y_i| \tag{5.54}$$

$$\delta_{\cos}(x, y) = \frac{-x^T y}{\|x\|\|y\|} \tag{5.55}$$

The proposed algorithm comprises the following steps.

Step 1. Use the Gabor wavelet analysis to transform the original image $I(x, y)$ into the Gabor feature vector $F, (F \in R^m)$.

Step 2. Use KPCA with the FPP models, $k(x, y) = (x \cdot y)^d$, $(0 < d < 1)$, to transform the m dimensional Gabor feature space R^m into a n dimensional space R^n.

Step 3. Based on LDA, in R^n, we construct the between-class S_b and within-class S_w, and then calculate the orthonormal eigenvectors, $\alpha_1, \ldots, \alpha_n$ of S_w.

Step 4. Extract the regular discriminant features as follows. Let $P_1 = (\alpha_1, \ldots, \alpha_q)$, where $\alpha_1, \ldots, \alpha_q$ corresponds to q positive eigenvalues and $q = \text{rank}(S_w)$. Calculate the eigenvectors $\beta_1, \ldots, \beta_l$, $(l \leq c - 1)$of $\widetilde{S}_b \zeta = \lambda \widetilde{S}_w \zeta$ corresponding to l largest positive eigenvalues, where $\widetilde{S}_b = P_1^T S_b P_1$, $\widetilde{S}_w = P_1^T S_w P_1$, and c is the number of face classes. We can calculate the regular discriminant feature vector $f_{\text{regular}} = B^T P_1^T y$, where $y \in R^n$ and $B = (\beta_1, \ldots, \beta_l)$.

Step 5. Extract the irregular discriminant features as follows. Calculate the eigenvectors $\gamma_1, \gamma_2, \ldots, \gamma_l$, $(l \leq c - 1)$ of $\widetilde{S}_b$ corresponding to l largest positive eigenvalues. Let $P_2 = (\alpha_{q+1}, \ldots, \alpha_m)$, the regular discriminant feature vector $f_{\text{irregular}}$ can be calculated by $f_{\text{irregular}} = C^T P_2^T y$, where $C = (\gamma_1, \ldots, \gamma_l)$ and $y \in R^n$.

Step 6. Fuse the irregular discriminant feature $f_{\text{irregular}}$ and the regular discriminant feature f_{regular} with the proper measure. Let $S(f_x, f_y)$ denotes the similarity between the two feature vectors, and κ denotes the fusion coefficient. $S(f_x, f_y)$ is calculated by fusing the two feature information with $S(f_x, f_y) = \kappa \cdot S(f_{\text{xirregular}}, f_{\text{yirregular}}) + S(f_{\text{xregular}}, f_{\text{yregular}})$.

Step 7. In the feature space, classify a new face F_{new} into the class with the closest mean C_k based on the similarity measure $S(F_{\text{new}}, C_k) = \min_j S(F_{\text{new}}, C_j)$.

5.6 Nonparametric Kernel Discriminant Analysis

5.6.1 Motivation

Among these classifiers, LDA method is the popular and effective method. LDA as the dimensionality reduction method is widely used in many areas. However, most LDA-based methods should satisfy two preconditions, i.e., the unimodal distribution of samples and the different scatter of class means of samples, but it is difficult to satisfy these preconditions in the practical applications. LDA often encounters the so-called ill-posed problems when applied to a small samples size problem. Recently, Li et al. [33] proposed nonparametric discriminant analysis (NDA) reported an excellent recognition performance for face recognition. However, NDA has its limitations on extracting the nonlinear features of face images for recognition because the distribution of images, such as face images, under a perceivable variation in viewpoint, illumination or facial expression is highly nonlinear and complex, and the linear techniques cannot provide reliable and robust solutions to those face recognition problems with complex face variations.

The detail algorithm is shown as follows:

Supposed that

$$S_W = \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} \left(x_i^l - N(x_i^l, i, k)\right)\left(x_i^l - N(x_i^l, i, k)\right)^T \tag{5.56}$$

$$S_B = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j \neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w(i,j,k,l)\left(x_i^l - N(x_i^l, j, k)\right)\left(x_i^l - N(x_i^l, j, k)\right)^T \tag{5.57}$$

where $w(i,j,k,l)$ is defined as

$$w(i,j,k,l) = \frac{\min\left\{d^\alpha\left(x_i^l, N\left(x_i^l, i, k\right)\right), d^\alpha\left(x_i^l, N\left(x_i^l, j, k\right)\right)\right\}}{d^\alpha(x_i^l, N(x_i^l, i, k)) + d^\alpha(x_i^l, N(x_i^l, j, k))} \tag{5.58}$$

where $d(v_1, v_2)$ is the Euclidean distance between two vector v_1 andv_2, and α is the a parameter ranging from zero to infinity which control the changing of the weight with respect to the ratio of the distance. $N\left(x_i^l, j, k\right)$ is the kth nearest neighbor from class j to the vector x_i^l which from lth sample of ith class.

Algorithm Procedure:

Step 1. Calculate kth nearest neighbor vector $N\left(x_i^l, j, k\right)$ from class j to the vector x_i^l which from lth sample of ith class.

Step 2. Calculate the Euclidean distance $d(v_1, v_2)$ between two vector v_1 andv_2, and then calculate $w(i,j,k,l) = \frac{\min\left\{d^\alpha\left(x_i^l, N\left(x_i^l, i, k\right)\right), d^\alpha\left(x_i^l, N\left(x_i^l, j, k\right)\right)\right\}}{d^\alpha\left(x_i^l, N\left(x_i^l, i, k\right)\right) + d^\alpha\left(x_i^l, N\left(x_i^l, j, k\right)\right)}$.

Step 3. Calculate $S_W = \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} \left(x_i^l - N(x_i^l, i, k)\right)\left(x_i^l - N(x_i^l, i, k)\right)^T$.

Step 4. Calculate $S_B = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j\neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w(i,j,k,l)\left(x_i^l - N(x_i^l, j, k)\right)\left(x_i^l - N(x_i^l, j, k)\right)^T$.

Step 5. Calculate the projection matrix W through solving the eigenvector $S_W^{-1}S_B$ corresponding to the largest eigenvalue.

5.6.2 Method

NDA algorithm is essentially a linear feature extraction algorithm. Although it is reported good performance on face recognition, however, there still is the space to improve its recognition performance through enhancing its ability to extracting the nonlinear facial features owing to the variation in illumination. In order to solve this limitation, we introduce the kernel trick develop a novel feature method called Nonparametric Kernel Discriminate Analysis (NKDA) as follows. The main idea is based on a conceptual transformation from the input space into a nonlinear high-dimensional feature space. Supposed that M training samples $\{x_1, x_2, \ldots, x_M\}$ with L class labels take values in an N-dimensional space $\mathbb{R}^N$, the data in $\mathbb{R}^N$ are mapped into a feature space F via the following nonlinear mapping $\Phi : \mathbb{R}^N \to F,\ x \mapsto \Phi(x)$. The main idea of kernel based method is to map the input data into a feature space by a nonlinear mapping where inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly. The nonlinear mapping is referred to as the "empirical kernel map", and the nonlinear mapping space is called "empirical feature space". Three kinds of kernel functions used in the kernel based learning machine are polynomial kernels, Gaussian kernels, and sigmoid kernels.

We have theoretical analysis on NDA in the feature space F to develop NKDA. Supposed that

$$S_W^{\Phi} = \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} \left(\Phi(x_i^l) - N(\Phi(x_i^l), i, k)\right)\left(\Phi(x_i^l) - N(\Phi(x_i^l), i, k)\right)^T \tag{5.59}$$

$$S_B^{\Phi} = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j\neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w^{\Phi}(i,j,k,l)\left(\Phi(x_i^l) - N(\Phi(x_i^l), j, k)\right)\left(\Phi(x_i^l) - N(\Phi(x_i^l), j, k)\right)^T \tag{5.60}$$

where $w^{\Phi}(i,j,k,l)$ is defined as

$$w^{\Phi}(i,j,k,l) = \frac{\min\left\{d^{\alpha}\left(\Phi(x_i^l), N\left(\Phi(x_i^l), i, k\right)\right), d^{\alpha}\left(\Phi(x_i^l), N\left(\Phi(x_i^l), j, k\right)\right)\right\}}{d^{\alpha}(\Phi(x_i^l), N(\Phi(x_i^l), i, k)) + d^{\alpha}(\Phi(x_i^l), N(\Phi(x_i^l), j, k))} \tag{5.61}$$

where $d(\Phi(v_1), \Phi(v_2))$ is the Euclidean distance between two vector v_1 and v_2 in the kernel space, and α is the a parameter ranging from zero to infinity which control the changing of the weight with respect to the ratio of the distance. $N\left(\Phi\left(x_i^l\right), i, k\right)$ is the kth nearest neighbor from class j to the vector x_i^l which from lth sample of ith class with the similarity measure with Euclidean distance in the kernel space.

In order to the clear description of NKDA, we rewrite the equation as

$$S_W^{\Phi} = \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} \left(\Phi\left(x_i^l\right) - \Phi\left(y_i^k\right)\right)\left(\Phi\left(x_i^l\right) - \Phi\left(y_i^k\right)\right)^T \tag{5.62}$$

$$S_B^{\Phi} = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j \neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w^{\Phi}(i,j,k,l)\left(\Phi\left(x_i^l\right) - \Phi\left(y_j^k\right)\right)\left(\Phi\left(x_i^l\right) - \Phi\left(y_j^k\right)\right)^T \tag{5.63}$$

$$w^{\Phi}(i,j,k,l) = \frac{\min\left\{d^{\alpha}\left(\Phi\left(x_i^l\right), \Phi\left(y_i^k\right)\right), d^{\alpha}\left(\Phi\left(x_i^l\right), \Phi\left(y_i^k\right)\right)\right\}}{d^{\alpha}\left(\Phi(x_i^l), \Phi\left(y_j^k\right)\right) + d^{\alpha}\left(\Phi(x_i^l), \Phi\left(y_j^k\right)\right)} \tag{5.64}$$

where $\Phi\left(y_i^k\right) = N\left(\Phi\left(x_i^l\right), i, k\right)$ and $\Phi\left(y_j^k\right) = N\left(\Phi\left(x_i^l\right), j, k\right)$. It is easy to obtain

$$d^{\alpha}\left(\Phi\left(x_i^l\right), \Phi\left(y_i^k\right)\right) = \left\|\Phi\left(x_i^l\right) - \Phi\left(y_i^k\right)\right\|^{\alpha} = \left(k\left(x_i^l, x_i^l\right) - 2k\left(x_i^l, y_i^k\right) + k\left(y_i^k, y_i^k\right)\right)^{\frac{\alpha}{2}} \tag{5.65}$$

FC is defined by

$$J(V) = \frac{V^T S_B^{\Phi} V}{V^T S_W^{\Phi} V} \tag{5.66}$$

where V is the discriminant vector, and S_B^{Φ} and S_W^{Φ} are the between classes scatter matrix and the total population scatter matrix respectively. According to the Mercer kernel function theory, any solution V belongs to the span of all training pattern in $\mathbb{R}^N$. Hence there exist coefficients $c_p\,(p = 1, 2, \ldots, M)$ such that

$$V = \sum_{p=1}^{M} c_p \Phi(x_p) = \Psi\alpha \tag{5.67}$$

where $\Psi = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_M)]$ and $\alpha = [c_1, c_2, \ldots, c_M]^T$.

$$J(\alpha) = \frac{\alpha^T B \alpha}{\alpha^T W \alpha} \tag{5.68}$$

where $B = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j \neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w^{\Phi}(i,j,k,l) B(i,j,k,l)$ and $W = \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} W(i,k,l)$. The proof of Eq. (5.68) is shown as follows.

$$
\begin{aligned}
J(\alpha) &= \frac{\alpha^T \Psi^T \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j\neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w^{\Phi}(i,j,k,l)\left(\Phi\left(x_i^l\right) - \Phi\left(y_j^k\right)\right)\left(\Phi\left(x_i^l\right) - \Phi\left(y_j^k\right)\right)^T \Psi\alpha}{\alpha^T \Psi^T \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} \left(\Phi\left(x_i^l\right) - \Phi\left(y_i^k\right)\right)\left(\Phi\left(x_i^l\right)^T - \Phi\left(y_i^k\right)^T\right)\Psi\alpha} \\
&= \frac{\alpha^T \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j\neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w^{\Phi}(i,j,k,l) B(i,j,k,l)\alpha}{\alpha^T \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} W(i,k,l)\alpha} \\
&= \frac{\alpha^T B\alpha}{\alpha^T W\alpha}
\end{aligned}
\tag{5.69}
$$

where $B = \sum_{i=1}^{c} \sum_{\substack{j=1 \\ j\neq i}}^{c} \sum_{k=1}^{k_2} \sum_{l=1}^{n_i} w^{\Phi}(i,j,k,l) B(i,j,k,l)$ and $W = \sum_{i=1}^{c} \sum_{k=1}^{k_1} \sum_{l=1}^{n_i} W(i,k,l)$.where $B(i,j,k,l) = K_1(i,j,k,l)^T K_1(i,j,k,l)$, $W(i,k,l) = K_2(i,k,l)^T K_2(i,k,l)$, $K_1(i,j,k,l) = \left[k\left(x_1, x_i^l\right), \ldots, k\left(x_M, x_i^l\right)\right] - \left[k\left(x_1, y_j^k\right), \ldots, k\left(x_M, y_j^k\right)\right]$ and $K_2(i,k,l) = \left[k\left(x_1, x_i^l\right), \ldots, k\left(x_M, x_i^l\right)\right] - \left[k\left(x_1, y_i^k\right), \ldots, k\left(x_M, y_i^k\right)\right]$where $B(i,j,k,l) = K_1(i,j,k,l)^T K_1(i,j,k,l)$, $W(i,k,l) = K_2(i,k,l)^T K_2(i,k,l)$, $K_1(i,j,k,l) = \left[k\left(x_1, x_i^l\right), \ldots, k\left(x_M, x_i^l\right)\right] - \left[k\left(x_1, y_j^k\right), \ldots, k\left(x_M, y_j^k\right)\right]$ and $K_2(i,k,l) = \left[k\left(x_1, x_i^l\right), \ldots, k\left(x_M, x_i^l\right)\right] - \left[k\left(x_1, y_i^k\right), \ldots, k\left(x_M, y_i^k\right)\right]$.

The projection matrix $V = [\alpha_1, \alpha_2, \ldots, \alpha_d]$ is easy to be obtained by the eigenvectors of $W^{-1}B$ corresponding to the d largest eigenvalue. Calculate the eigenvectors of $W^{-1}B$ is the same to simultaneous diagonalization of W and B. Firstly, the eigenvector matrix Φ and the corresponding eigenvalue matrix Θ of W are solved, and then project the class centers onto $\Phi\Theta^{-1/2}$, thus B is transformed to $B_K = \Theta^{-1/2}\Phi^T B \Phi\Theta^{-1/2}$. Finally, solve the eigenvector matrix Ψ and the eigenvalue matrix Λ of B_K, the projection matrix V is equal to $V = \Phi\Theta^{-1/2}\Psi$.

5.7 Experiments on Face Recognition

5.7.1 Experimental Setting

We construct the experimental system as shown in Fig. 5.1 for evaluation algorithms. After describing the two data set used in our experiments. It is worthwhile to make some remarks on the experiment setting as follows. (1) We randomly select five images from each subject from ORL face database for training, and the

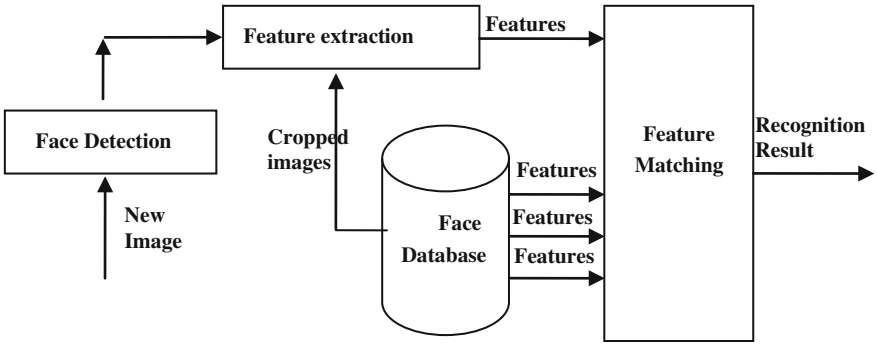


Fig. 5.1 Framework of a real system example

rest images are used to test the performance. Similarly, five images of each person randomly selected from YALE database are used to construct the training set, and the rest five images of each person are used to test the performance of the algorithms. (2) We run each set of experiments for 10 times, and the averaged results are used to evaluate the performance of the proposed algorithm. (3) The experiments are implemented on a Pentium 3.0 GHz computer with 512 MB RAM and programmed in the MATLAB platform (Version 6.5). (4) The procedural parameters are chosen with cross-validation method.

5.7.2 Experimental Results of AQKDA

The proposed methods are implemented on ORL and YALE databases. Firstly we evaluate the proposed four methods of choosing XVs and two criterions of solving the expansion coefficients, and secondly we test the superiority of AQKDA compared with KDA, KPCA and KWMMDA.

Firstly, we evaluate four methods of choosing XVs and two criterions of solving expansion coefficients [i.e., FC and MMC]. Tables 5.3 and 5.4 show the performance on recognition accuracy and computation efficiency. MMC achieves a higher recognition accuracy and computation efficiency compared with FC. We evaluate the computation efficiency of FC and MMC through comparing the time consuming of solving β using FC with the time consuming of solving β using MMC. The four methods of choosing XVs achieve the approximately same recognition accuracy but the different computation efficiency. The dimension of the matrix E corresponding to the method of choosing XVs have great impact on the computation efficiency of solving the expansion coefficients. The dimension of the matrix E corresponding to XVs 3 and XVs 4 is smaller than one of the matrix E corresponding to XVs 1 and XVs 2. As shown in Tables 5.2 and 5.3, XVs 3 and

Table 5.2 MMC versus Fisher with four methods of choosing XVs on ORL face database

		XVs 1	XVs 2	XVs 3	XVs 4
Recognition accuracy	MMC	0.9365	0.9340	0.9335	0.9355
	FC	0.9245	0.9215	0.9210	0.9240
Time consuming (s)	MMC	0.05	0.05	0.02	0.02
	FC	1.50	1.48	0.20	0.19

Table 5.3 MMC versus Fisher with four methods of choosing XVs on YALE face database

		XVs 1	XVs 2	XVs 3	XVs 4
Recognition accuracy	MMC	0.9187	0.9227	0.9200	0.9227
	FC	0.9000	0.9147	0.9079	0.9187
Time consuming (s)	MMC	0.03	0.05	0.02	0.02
	FC	0.32	0.32	0.08	0.06

Table 5.4 Performance comparisons of AQKDA, KDA, KPCA and KWMMDA on ORL and YALE face databases

	ORL database	YALE database
AQKDA	0.9410	0.9267
KDA	0.9250	0.9040
KPCA	0.8180	0.8253
KWMMDA	0.7980	0.7653

XVs 4 outperform XVs 1 and XVs 2 on the computation efficiency. In the next experiments, we applied the MMC to solve the expansion coefficients.

Secondly, we compare the classification performance of AQKDA, KDA, KPCA and KWMMDA on the two datasets. All the procedural parameters of these algorithms are chosen with cross-validation methods. As results shown in Table 5.4, AQKDA achieves the highest recognition rate compared with other algorithms. The data in the feature space has the largest class discrimination via adaptive quasiconformal kernel mapping, so AQKDA outperforms KDA on ORL and YALE datasets. Although the procedure of finding adaptive expansion coefficients vector incurs a small fractional computational overhead, it does not cause significantly increasing of the response time, which can be ignored by comparing with the improved performance.

The experiments on the two real-world datasets have been systematically performed. These experiments reveal a number of interesting points: (1) All the experiments indicate that AQKDA consistently performs better than KDA and other algorithms. It also indicates that AQKDA is adaptive to the input data for classification. (2) In the experiments, the four methods of choosing the XVs achieve the same recognition accuracy at most, while perform differently on the computation efficiency. Since the dimensionality of the matrix E is reduced with the method XVs 3 and XVs 4, the two methods perform better on the time consuming. So if the number of samples per class is very large while the number of class is not so large, the method XVs 3 and XVs 4 are expected to perform more

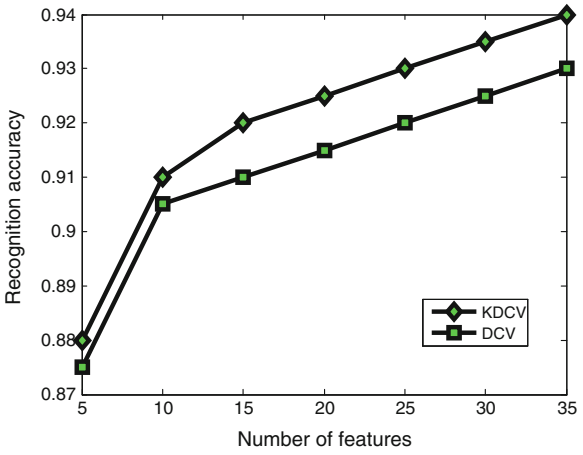


Fig. 5.2 Performance comparison of DCV and its kernel version on ORL face database

advantage than other two methods. (3) Two criterions of solving the expansion coefficients perform differently both on recognition accuracy and computation efficiency. MMC obtains a better performance than FC.

5.7.3 Experimental Results of Common Kernel Discriminant Analysis

In the experiments, we test the feasibility of improving the recognition performance using Gabor feature analysis and common vector analysis, and then we evaluate the recognition performance of CGV on the real datasets. The

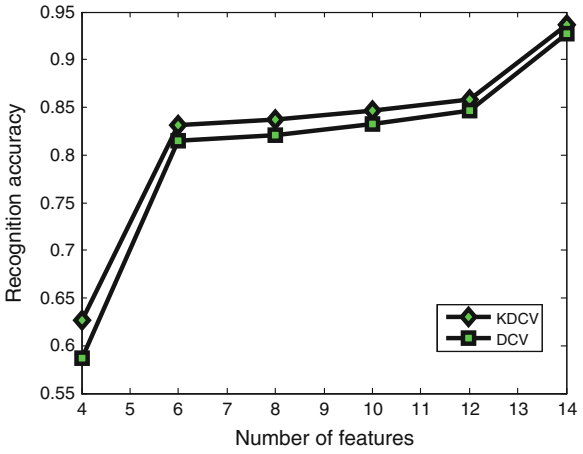


Fig. 5.3 Performance comparison of DCV and its kernel version on Yale face database

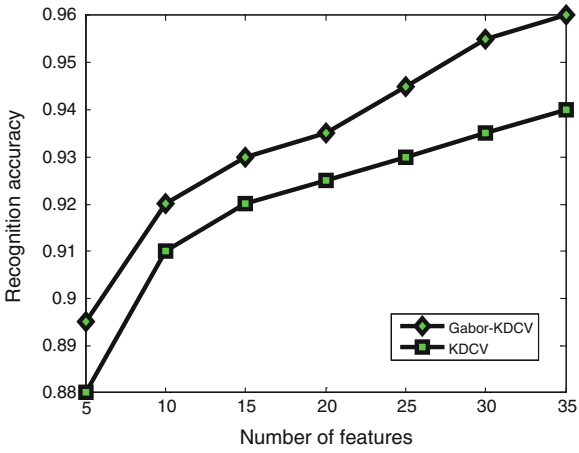


Fig. 5.4 Evaluation of the feasibility of enhancing the recognition performance with Gabor features analysis on ORL database

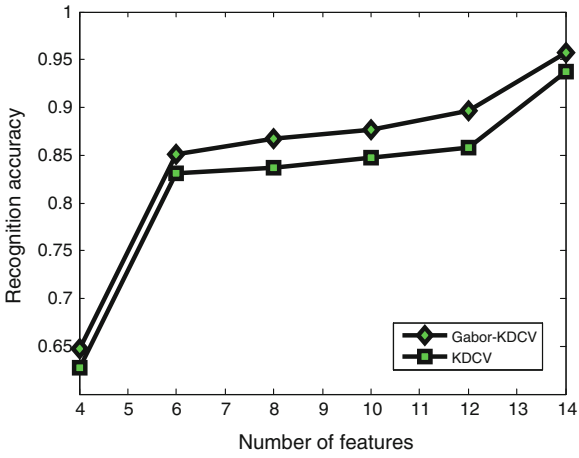


Fig. 5.5 Evaluation of the feasibility of enhancing the recognition performance with Gabor features analysis on Yale database

effectiveness is evaluated with both absolute performance and comparative performance against other current methods such as PCA, KPCA, LDA, KFD.

The feasibility of kernel trick and Gabor analysis are testified respectively. As shown in Figs. 5.2 and 5.3, KDCV outperforms DCV under the same dimensionality of feature vector, which demonstrates that kernel method is effective to enhance the recognition performance of DCV. As shown in Figs. 5.4 and 5.5, Gabor features analysis method improves the recognition accuracy on the same experimental conditions on the two databases, where the parameter $S = 2,\ K = 4$ is chosen for Gabor wavelet. Since the face images from ORL and Yale databases are obtained with the different illuminations, poses and expressions conditions, and the good performance shows that kernel method and Gabor feature analysis are feasible to solve this PIE problem of face recognition. We evaluate CGV compared

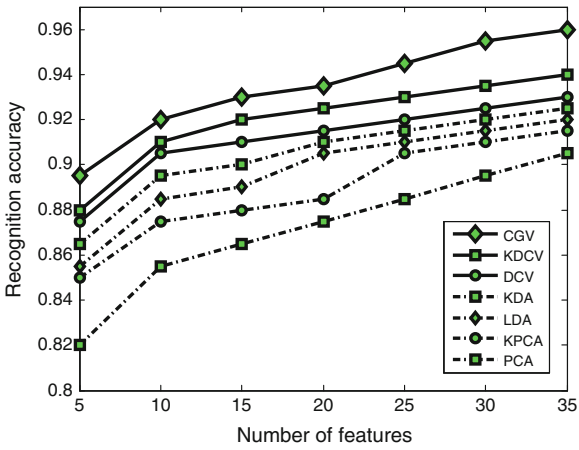


Fig. 5.6 Recognition performance of CGV on ORL database compared with KDCV, DCV, KDA, LDA, KPCA, PCA

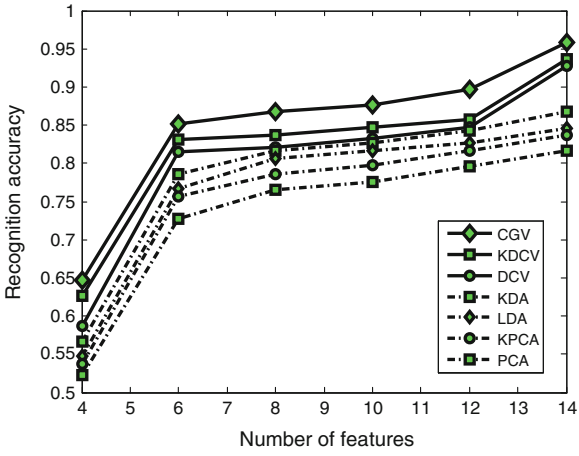


Fig. 5.7 Recognition performance of CGV on Yale database compared with KDCV, DCV, KDA, LDA, KPCA, PCA

with PCA, KPCA, LDA, KDA, DCV, KDCV on the recognition performance, and recognition results are shown in Figs. 5.6 and 5.7. CGV method achieves a highest recognition accuracy compared with the other algorithms. The face images from two face databases are achieved under the different poses, illuminations and expressions conditions, and it demonstrates that CGV method is more robust to the change in pose, illumination, expression for face recognition compared with other methods.

5.7.4 Experimental Results of CKFD

There are many parameters needs to be decided through cross-validation method. It is not possible to decide all the parameters at one time, we only set some values in advance and then select other parameters through cross-validation method. In the simulation, we select the Euclidean distance measure as the similarity measure and $S = 2,\ K = 4$ as parameters of Gabor wavelet for Gabor feature extraction. Through cross-validation method, the fusion coefficient is set to $\kappa = 0.3$ for fusion of two kinds of discriminant feature information in CKFD. Then we select the procedural parameters through cross-validation method including the kernel parameters (i.e., the order d of polynomial kernels and the width σ^2 of Gaussian kernels) and the degree d of FPP models for recognition performance, fusion coefficient and similarity measures (i.e. Euclidean distance, $L1$ distance, and cosine similarity measures). Six different orders ($d = 1,\ d = 2,\ d = 3,\ d = 4,\ d = 5$, and $d = 6$) of polynomial kernel are set for $k(x, y) = (x \cdot y)^d, (d \in N)$. The polynomial kernel with the first order ($d = 1$) performs the best, followed by the other orders of polynomial kernels ($d = 2,\ d = 3,\ d = 4,\ d = 5$, and $d = 6$). We find that the lower the order of polynomial kernel is, the better the recognition performance is. It seems that the performance of FPP models with $d < 1$ is better than the polynomial kernel with the first order $d = 1$. For testifying it, we implement the simulations with FPP models $k(x, y) = (x \cdot y)^d\ (0 < d < 1)$ [25]. As shown in Table 5.5, recognition performance performs best with the degree $d = 0.8$ of FPP models. For testing the superiority of FPP models in the next step, we also assess the performance of Gaussian kernels $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$ with different width σ^2 to select the optimal parameter, the width $\sigma^2 = 0.8 \times 10^6$ for Gaussian kernels performs best.

We select the similarity measures (i.e. Euclidean distance measure, $L1$ distance measure and cosine similarity measure) through cross-validation method. As shown in Table 5.6, the Euclidean distance performs better than $L1$ distance

Table 5.5 Procedural parameters selection through cross-validation method

Polynomial kernel	d	1	2	3	4	5	6
	Accuracy	0.965	0.905	0.785	0.745	0.685	0.630
FPP model	d	0.5	0.6	0.7	0.8	0.9	0.95
	Accuracy	0.935	0.940	0.965	0.970	0.960	0.965
Gaussian kernel	σ^2	0.2×10^6	0.4×10^6	0.6×10^6	0.8×10^6	1.0×10^6	1.2×10^6
	Accuracy	0.915	0.945	0.955	0.965	0.965	0.965

Table 5.6 Recognition rates versus the number of features, using NNC with Euclidean distance

Feature dimension	5	10	15	20	25	30	35
Euclidean distance	0.905	0.925	0.955	0.960	0.965	0.970	0.980
$L1$ distance	0.89	0.910	0.940	0.950	0.955	0.965	0.970
Cosine similarity measure	0.87	0.90	0.93	0.935	0.940	0.945	0.950

Table 5.7 Performance comparison of FPP model, polynomial and Gaussian kernel

Feature dimension	5	10	15	20	25	30	35
FPP models	0.905	0.925	0.955	0.960	0.965	0.970	0.980
Polynomial kernels	0.805	0.895	0.915	0.925	0.935	0.940	0.945
Gaussian kernels	0.785	0.880	0.895	0.90	0.915	0.930	0.935

measure and cosine similarity measure. So the Euclidean distance measure is chosen as the similarity measure.

In the procedural parameters selection part, the optimal order $d = 1$ for polynomial kernels, the optimal width $\sigma^2 = 0.8 \times 10^6$ for Gaussian kernels, the optimal degree $d = 0.8$ of the FPP models are chosen through cross-validation method. In the following the simulations, we test the feasibility of improving performance using FPP model and Gabor wavelet. Firstly, we compare the performance of FPP models with other two kinds of kernels with optimal parameters. As shown in Table 5.7, Gabor-based CKFD with FPP models method performs best. That is, enhancing the recognition performance using FPP models is feasible.

After testing the feasibility of enhancing recognition performance using FPP models, we test the feasibility of improving the recognition performance using Gabor wavelets in this part of simulations. We choose the proper Gabor wavelets parameters firstly, and then compare recognition performance of Gabor-based CKFD with FPP models with one of CKFD with FPP models. As shown in Table 5.8, Gabor wavelet with the number of scales $S = 2$ performs better than

Table 5.8 Recognition rates versus the number of features, using NNC with Euclidean distance

Feature dimension	5	10	15	20	25	30	35
Gabor(2,4)	0.905	0.925	0.945	0.960	0.965	0.970	0.980
Gabor(4,4)	0.87	0.89	0.91	0.925	0.935	0.94	0.9450

Table 5.9 Recognition performance of Gabor-based CKFD with FPP models versus CKFD with FPP model

Feature dimension	5	10	15	20	25	30	35
Gabor-CKFD with FPP models	0.905	0.925	0.945	0.960	0.97	0.975	0.980
CKFD with FPP models	0.89	0.91	0.925	0.94	0.945	0.95	0.9550

Table 5.10 Performance on ORL face database

Feature dimension	5	10	15	20	25	30	35
Proposed	0.905	0.925	0.945	0.970	0.97	0.975	0.980
CKFD	0.885	0.915	0.925	0.940	0.945	0.95	0.9550
KFD	0.870	0.890	0.905	0.925	0.930	0.935	0.940
LDA	0.865	0.885	0.890	0.905	0.915	0.925	0.930
KPCA	0.840	0.860	0.875	0.885	0.895	0.905	0.915
PCA	0.825	0.845	0.855	0.86	0.875	0.89	0.905

with the number of scales $S = 4$. As the results shown in Table 5.9, the Gabor-based CKFD with FPP models method performs better than the CKFD with FPP models method. It also indicates that the Gabor wavelets can enhance the face recognition performance.

We implement Gabor-based CKFD with FPP models method with two face databases. We also implement the popular subspace face recognition methods, i.e., principal component analysis (PCA), Linear (Fisher) discriminant analysis (LDA), KPCA, KFD and CKFD.

The algorithms are implemented in the ORL face database, and the results are shown in Table 5.10, which indicate that our method performs better than other popular method. And we can obtain the accuracy rate 0.98 with our method. Especially we can acquire the peak recognition rate 0.97 when the number of feature is only 35, and we can also obtain the recognition rate 0.905 even if the number of feature decreases to five. Since the feature number of the face image will influence the computational cost for face recognition, the proposed method using the small feature number performs better than the PCA, LDA, KPCA, KFD, and CKFD methods. The proposed method improves the computational efficiency without influencing the recognition performance. Because in the ORL face database, the variations of the images are across pose, time and facial expression, the simulations also indicate our method is robust to the change in poses and expressions.

The same comparison results are obtained with Yale and UMIST face database shown in Tables 5.11 and 5.12. As shown as Table 5.11, the peak recognition rate with our method is 0.96 that is higher than other methods, which indicates that our method is robust to the change of illumination and expression. The highest accuracy of 0.945 is obtained on UMIST face database.

From the above simulations, we can also acquire the following interesting points:

All the face recognition approaches mentioned in the simulations perform better in the optimal face subspace than in the original image space.

Among all the face recognition methods used in the simulations, kernel-based methods perform better than methods based on the linear subspace. It also indicates the kernel-based method is promising in the area of face recognition.

Table 5.11 Performance on Yale face database

Methods	PCA	KPCA	LDA	KFD	CKFD	Our method
Accuracy rate	0.733	0.767	0.867	0.933	0.947	0.960

Table 5.12 Performance on UMIST face database

Methods	PCA	KPCA	LDA	KFD	CKFD	Our method
Accuracy rate	0.815	0.840	0.885	0.895	0.925	0.945

In the first part of simulations, Gabor-based CKFD with FPP models method performs better than with Gaussian kernels and polynomial kernels. Although a FPP model might not define a positive semi-definite Gram matrix, it has been successfully used in practice. As results shown in our simulations, FPP models enhance the face recognition performance.

The advantage of Gabor feature characterized by spatial frequency, spatial locality and orientation selectivity to cope with the variations due to illumination and facial expression changes, is approved in the second part of simulations. As the results shown in the simulations, Gabor wavelets improve the face recognition performance. What attracts our attention is that Gabor wavelets with low number of scales perform better than ones with high number of scales, while the choice of the number of orientations affects not so much the recognition performance. For the face images, the distribution of feature information mostly converges in the low frequency phase, so overfull high frequency information will influence representation of the discriminant features.

Selection of kernel functions and kernel parameters will influence the recognition performance straightly and the parameters are selected through cross-validation method. It is worth to study further that kernel parameters are automatically optimized.

We implement the simulation with ORL face database, Yale face database and UMIST face database. The face images of the three face databases are taken under different lighting conditions and different facial expressions and poses. In all simulations, our approach consistently performs better than the PCA, LDA, KPCA, KFD and CKFD approaches. Our method also provides a new idea to do with PIE problem of face recognition.

5.7.5 *Experimental Results of NKDA*

In our experiments, we implement our algorithm in the three face databases, ORL, YALE and UMIST face databases. In the experiments, we implement three parts of experiments to evaluate the performance on feature extraction and recognition. In the experiments, we randomly select five images from each subject, 200 images in total for training, and the rest 200 images are used to test the performance. We run each set of experiments for 10 times, and the averaged results are used to evaluate the performance of the proposed algorithm. The experiments are implemented on a Pentium 3.0 GHz computer with 512 MB RAM and programmed in the MATLAB platform. The procedural parameters are chosen with cross-validation method. The experiments are implemented to testify the feasibility of improving the NDA with kernel trick under the same conditions. We choose three sets of parameters, i.e., $k_1 = 2$, $k_2 = 2$, $k_1 = 2$, $k_2 = 3$ and $k_1 = 3$, $k_2 = 3$, for NDA and NKDA under the different number of features for face recognition. And then we compare the recognition performance of the NDA and NKDA algorithms under the same dimension of the feature vector on ORL face database. We use the

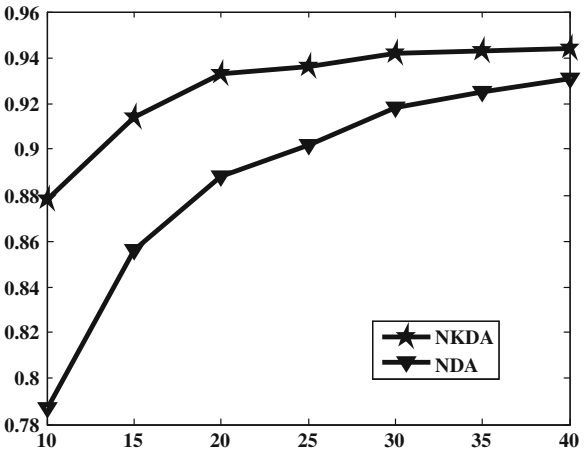


Fig. 5.8 NDA versus NKDA (k_1 =2, k_2 = 2) on ORL database

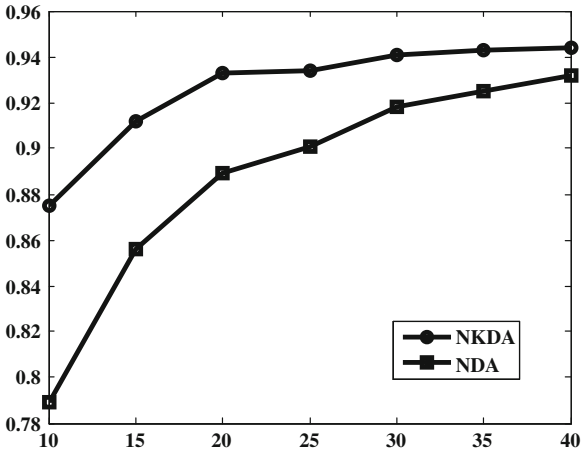


Fig. 5.9 NDA versus NKDA (k_1 = 2, k_2 = 3) on ORL database

recognition accuracy to evaluate the recognition performance of the two algorithms. The feasibility of kernel method is testified on the database through comparing the recognition performance of NKDA and NDA with the same procedural parameters.

1. Evaluation on ORL face database

In the experiments, we evaluate the recognition performance with the different feature dimension through comparing NKDA and NDA. Experimental results under the different procedural parameter are show in Figs. 5.8, 5.9 and 5.10.

As shown in in Figs. 5.3, 5.4, and 5.5, the proposed NKDA outperforms NDA under the same dimensionality. We compare their recognition performance on the same feature dimensionality from 10 to 40. As shown in Figs. 5.3, 5.4, and 5.5,

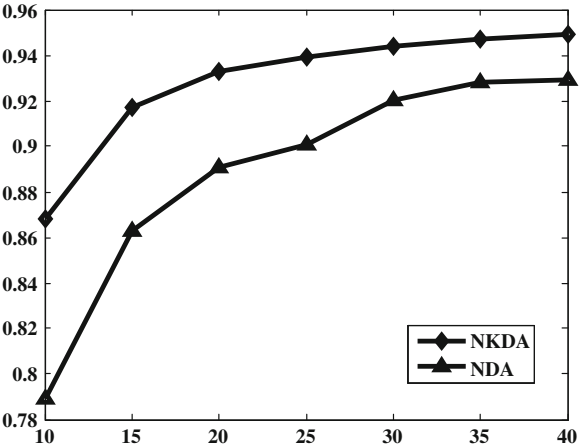


Fig. 5.10 NDA versus NKDA ($k_1 = 3$, $k_2 = 3$) on ORL database

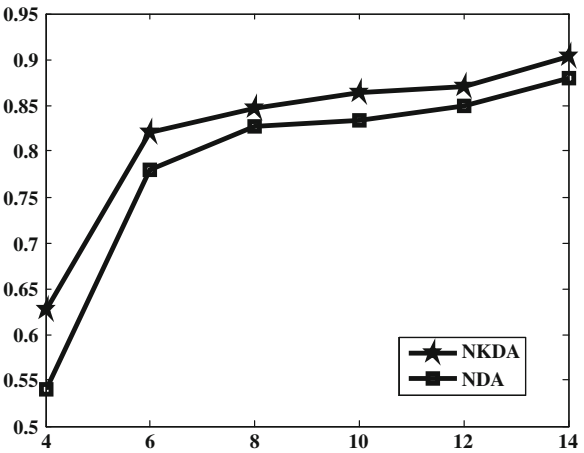


Fig. 5.11 NDA versus NKDA ($k_1 = 2$, $k_2 = 2$) on YALE database

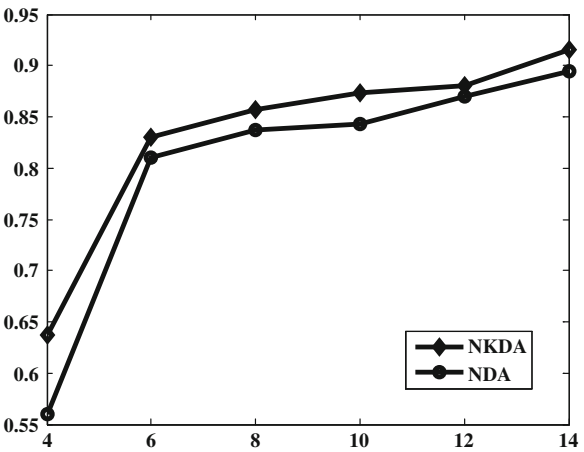


Fig. 5.12 NDA versus NKDA ($k_1 = 2$, $k_2 = 3$) on YALE database

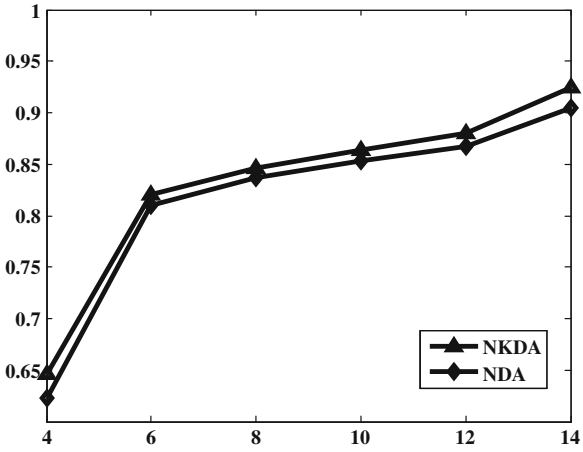


Fig. 5.13 NDA versus NKDA ($k_1 = 3$, $k_2 = 3$) on YALE database

under the same conditions, NKDA outperforms NDA on the recognition performance, which demonstrates that the kernel method is feasible to improve NDA. It is worth to emphasize that the recognition accuracy of NKDA is higher about 10 % than NDA with 10 features. NKDA achieves the higher recognition accuracy under the same feature dimension. So it is feasible to improve the performance of NDA with kernel method. Moreover, the NKDA algorithm is effective to deal with the illumination changing of face recognition, which is testified in the experiment.

2. Evaluation on YALE face database

The experimental results are shown in Figs. 5.11, 5.12 and 5.13. The same to the experiments implemented on YALE face database, the recognition accuracy is used to evaluate the performance of the proposed algorithm compared with linear

Fig. 5.14 Example face images of UMIST face database

Table 5.13 Performance on UMIST face database

Procedural parameters	NDA (%)	NKDA (%)
$k_1 = 2, k_2 = 2$	91.50	92.50
$k_1 = 2, k_2 = 3$	89.50	91.15
$k_1 = 3, k_2 = 3$	92.50	94.25

NDA algorithm. The different dimensionality of features is considered in the experiments. As shown in Figs. 5.7, 5.8 and 5.9, the NKDA achieves the higher recognition accuracy compared with NNDA algorithm under the same dimensionality of feature from 4 to 14, which shows that it is feasible to enhance the NDA with kernel trick.

3. Evaluation on UMIST face database

UMIST face database consists of 564 images of 20 people. Each covers a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. Each subject exists in their own directory labelled 1a, 1b,… 1t and images are numbered sequentially as they were taken. Some image examples are shown in Fig. 5.14.

In this part of experiments, we also use the recognition rate to evaluate the performance of NKDA and NDA algorithms, we use the highest recognition rate to evaluate the recognition performance. Experimental results shown in Table 5.13 show that the proposed NKDA algorithm outperforms NDA algorithm under the same conditions. So it is feasible to enhance NDA algorithm using kernel method to present the NKDA algorithm.

From the above experimental results, we can achieve the following points: (1) Among the mentioned face recognition methods, kernel-based nonlinear feature extraction methods perform better than other linear ones, which indicates kernel-based method is promising for face recognition. (2) Extracting the Gabor features of face image is feasible to deal with the variations in illumination and facial expression. The distribution of feature information mostly converges in the low frequency phase, but the overfull high frequency information influences representation of the discriminant features for face recognition. (3) Kernel functions and the corresponding parameters have high influence the recognition performance straightly, and the procedural parameters are chosen through cross-validation method, so how to choose them automatically should be researched in the future work.

References

1. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 12:181–201
2. Li J-B, Chu S-C, Pan J-S, Ho J-H (2007) Adaptive data-dependent matrix norm based Gaussian kernel for facial feature extraction. Int J Innovative Comput, Inf Control 3(5):1263–1272

3. Sahbi H (2007) Kernel PCA for similarity invariant shape recognition. Neurocomputing 70:3034–3045
4. Mika S, Ratsch G, Weston J, Schölkopf B, Muller KR (1999) Fisher discriminant analysis with kernels. In: Proceedings of IEEE international workshop on neural networks for signal processing IX, pp 41–48, Aug 1999
5. Tao D, Tang X, Li X, Wu X (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal Mach Intell 28(7):1088–1099
6. Schölkopf B, Smola A, Muller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
7. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Networks 14(1):117–226
8. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
9. Liang Z, Shi P (2005) Uncorrelated discriminant vectors using a kernel method. Pattern Recogn 38:307–310
10. Liang Z, Shi P (2004) Efficient algorithm for kernel discriminant anlaysis. Pattern Recogn 37(2):381–384
11. Liang Z, Shi P (2004) An efficient and effective method to solve kernel Fisher discriminant analysis. Neurocomputing 61:485–493
12. Yang J, Frangi AF, Yang J, Zhang D, Jin Z (2005) KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. IEEE Trans Pattern Anal Mach Intell 27(2):230–244
13. Yang MH (2002) Kernel eigenfaces vs. kernel fisherfaces: face recognition using kernel methods. In: Proceedings of 5th IEEE international conference on automatic face and gesture recognition, pp 215–220, May 2002
14. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Networks 14(1):117–126
15. Zheng W, Zou C, Zhao L (2005) Weighted maximum margin discriminant analysis with kernels. Neurocomputing 67:357–362
16. Huang J, Yuen PC, Chen WS, Lai JH (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. In: Proceedings of the 6th IEEE international conference on automatic face and gesture recognition 2004
17. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst, Man Cybern-Part B: Cybern 35(3):556–562
18. Chen WS, Yuen PC, Huang J, Dai D-Q (2005) Kernel machine-based one-parameter regularized fisher discriminant method for face recognition. IEEE Trans Syst, Man Cybern-Part B: Cybern 35(4):658–669
19. Liang Y, Li C, Gong W, Pan Y (2007) Uncorrelated linear discriminant analysis based on weighted pairwise Fisher criterion. Pattern Recogn 40:3606–3615
20. Zheng Y, Yang J, Yang J, Wu X (2006) A reformative kernel Fisher discriminant algorithm and its application to face recognition. Neurocomputing 69(13-15):1806–1810
21. Tao D, Tang X, Li X, Rui Y (2006) Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. IEEE Trans Multimedia 8(4):716–727
22. Xu Y, Zhang D, Jin Z, Li M, Yang J-Y (2006) A fast kernel-based nonlinear discriminant analysis for multi-class problems. Pattern Recogn 39(6):1026–1033
23. Saadi K, Talbot NLC, Cawley GC (2007) Optimally regularised kernel Fisher discriminant classification. Neural Networks 20(7):832–841
24. Yeung D-Y, Chang H, Dai G (2007) Learning the kernel matrix by maximizing a KFD-based class separability criterion. Pattern Recogn 40(7):2021–2028
25. Shen LL, Bai L, Fairhurst M (2007) Gabor wavelets and general discriminant analysis for face identification and verification. Image Vis Comput 25(5):553–563
26. Ma B, Qu H, Wong H (2007) Kernel clustering-based discriminant analysis. Pattern Recogn 40(1):324–327

27. Wu X-H, Zhou J-J (2006) Fuzzy discriminant analysis with kernel methods. Pattern Recogn 39(11):2236–2239
28. Liu Q, Lu H, Ma S (2004) Improving kernel Fisher discriminant analysis for face recognition. IEEE Trans Pattern Anal Mach Intell 14(1):42–49
29. Lanckriet G, Cristianini N, Bartlett P, Ghaoui LE, Jordan MI (2004) Learning the kernel matrix with semidefinte programming. J Mach Learn Res 5:27–72
30. Xiong H, Swamy MNS, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Networks 16(2):460–474
31. Peng J, Heisterkamp DR, Dai HK (2004) Adaptive quasiconformal kernel nearest neighbor classification. IEEE Trans Pattern Anal Mach Intell 26(5):656–661
32. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Network 12(6):783–789
33. Li Z, Lin D, Tang X (2009) Nonparametric discriminant analysis for face recognition. IEEE Trans Pattern Anal Mach Intell 31(4):755–761

Chapter 6
Kernel Manifold Learning-Based Face Recognition

6.1 Introduction

Feature extraction with dimensionality reduction is an important step and essential process in embedding data analysis [1–3]. Linear dimensionality reduction aims to develop a meaningful low-dimensional subspace in a high-dimensional input space such as PCA [4] and LDA [5]. LDA is to find the optimal projection matrix with Fisher criterion through considering the class labels, and PCA seeks to minimize the mean square error criterion. PCA is generalized to form the nonlinear curves such as principal curves [6] or principal surfaces [7]. Principal curves and surfaces are nonlinear generalizations of principal components and subspaces, respectively. The principal curves are essentially equivalent to self-organizing maps (SOM) [8]. With the extended SOM, ViSOM preserves directly the distance information on the map along with the topology [9], which represents the nonlinear data [10] and represents a discrete principal curve or surface through producing a smooth and graded mesh in the data space. Recently, researchers proposed other manifold algorithms such as Isomap [11], locally linear embedding (LLE) [12] and locality preserving projection (LPP) [13]. LPP projects easily any new data point in the reduced representation space through preserving the local structure and intrinsic geometry of the data space [14]. Many improved LPP algorithms were proposed in recent years. Zheng et al. used the class labels of data points to enhance its discriminant power in the low-dimensional mapping space to propose supervised locality preserving projection (SLPP) for face recognition [15]. However, LPP is not orthogonal, which makes it difficult to reconstruct the data, so researchers applied the class information to present orthogonal discriminant locality preserving projections (ODLPP) for face recognition through orthogonalizing the basis vectors of the face subspace [16]. Cai et al. proposed the orthogonal locality preserving projection (OLPP) to produce orthogonal basis functions with more power of preserving locality than LPP [17]. OLPP was reported to have more discriminating power than LPP. Yu et al. introduced a simple uncorrelated constraint into the objective function to present uncorrelated discriminant locality preserving projections (UDLPP) with the aim of preserving the within-class

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_6,

geometric structure but maximizing the between-class distance [18]. In order to improve the performance of LPP on the nonlinear feature extraction, researchers perform UDLPP in reproducing kernel Hilbert space to develop kernel UDLPP for face recognition and radar target recognition. Feng et al. presented an alternative formulation of kernel LPP (KLPP) to develop a framework of KPCA+LPP algorithm [19]. With the application of the kernel methods in many areas [20, 21], other researchers improved LPP with kernels in the previous works [22–24]. In recent research, locality preserving projection and its improved methods are used in many areas, such as object recognition [25, 26], face recognition [27–29]. For any special image-based applications, such as face recognition, researchers proposed 2D LPP which extracts directly the proper features from image matrices without transforming one matrix into one vector [30–32].

From the above survey of LPP, researchers improved LPP from the following three cases: (1) supervised LPP, which uses the class labels to guide the creation of the nearest neighbor graph; (2) kernel LPP, which uses kernel tricks to enhance the performance on the nonlinear feature extraction; (3) 2D LPP, which deals with the image matrix without transforming 2D image into 1D vector. Our aim is to analyze and improve kernel LPP. Although the current kernel LPP algorithms are reported an excellent performance, kernel function and its parameter have significant influence on feature extraction owing to the fact that the geometrical structure of the data in the kernel mapping space is determined totally by the kernel function. If an inappropriate kernel is used, the data points in the feature space may become worse. However, choosing the kernel parameters from a set of discrete values will not change the geometrical structures of the data in the kernel mapping space. On the improved locality preserving projection and kernel optimization methods, firstly the work in [33] applies kernel optimization method to improve the performance of kernel discriminant analysis, while this book improves the traditional locality preserving projection with kernel optimization. Secondly, the work in [24] improves the traditional locality preserving projection with kernel method but without kernel optimization, so the algorithm is not able adaptively to change the kernel structure according to the input data. If the kernel function and its parameter are not chosen, the performance of locality preserving projection with kernel is reduced for feature extraction and recognition.

In this chapter, we propose a novel supervised feature extraction method namely kernel self-optimized locality preserving discriminant analysis (KSLPDA) based on the original definition of LPP to enhance the class structure of the data for classification. Different from LPP, KSLPDA guides the procedure of modeling the manifold with the class information and uses kernel self-optimization to overcome the kernel function selection problem endured by the traditional kernel LPP. In KSLPDA, the optimal local structure of the original data is achieved with the special consideration of both the local structure and the class labels with kernel self-optimization. The data-dependent kernel-based similarity has several good properties to discover the true intrinsic structure of data to make KSLPDA robust for classification tasks.

6.2 Locality Preserving Projection

Locality preserving projection (LPP) is a recently proposed dimensionality reduction method. The main idea of LPP is generally described as follows. Given a set of N-dimensional data $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^N$ from the L classes, LPP aims to find a transformation matrix W to map these points to a set of points $\{z_1, z_2, \ldots, z_n\}$ in $\mathbb{R}^d$ ($d \ll N$) based on one objective function z_i ($i = 1, 2, \ldots, n$). Different dimensionality reduction method has different objective function which is designed based on a different criterion. For example, LDA is based on Fisher criterion, and PCA aims to maximize the variance. The objective function of LPP is defined as follows:

$$\min \sum_{i,j} \left\| z_i - z_j \right\|^2 S_{ij} \tag{6.1}$$

where S is a similarity matrix with weights characterizing the likelihood of two points and z_i is the one-dimensional representation of x_i with a projection vector w, i.e., $z_i = w^T x_i$. By minimizing the objective function, LPP incurs a heavy penalty if neighboring mapped points z_i and z_j are far. LPP aims to keep the mapped points close where their original points are close each other. Thereof, the local information in the original space is preserved after dimension reduction with LPP. The local structure of data in the original space is preserved in lower-dimensional space. Minimizing (6.1) is equivalent to minimizing the following equation:

$$\frac{1}{2} \sum_{i,j} \left\| z_i - z_j \right\|^2 S_{ij} = w^T X (D - S) X^T w = w^T \mathrm{XLX}^T w \tag{6.2}$$

where $X = \{x_1, x_2, \ldots, x_n\}$, D is a diagonal matrix with its entries being the column or row (S is symmetric) sums of S, i.e., $D = \mathrm{diag}\left[\sum_j S_{1j}, \sum_j S_{2j}, \ldots, \sum_j S_{nj}\right]$, and $L = D - S$ is the Laplacian matrix. Matrix D presents local structural information of the data in the original space. The bigger the D_{ii} corresponding to x_i is, the more "important" is x_i. Impose a constraint as follows:

$$z^T D z = 1 \Rightarrow w^T \mathrm{XDX}^T w = 1 \tag{6.3}$$

Then, minimization problem reduces to

$$\arg\min_w w^T \mathrm{XLX}^T w$$

$$\text{Subject to } w^T \mathrm{XDX}^T w = 1 \tag{6.4}$$

The optimal transformation vector w is calculated through solving the following eigenvalue problem:

$$\mathrm{XLX}^T w = \lambda \mathrm{XDX}^T w \tag{6.5}$$

where $L = D - S$ and $D = \mathrm{diag}\left[\sum_j S_{1j}, \sum_j S_{2j}, \ldots, \sum_j S_{nj}\right]$. The similarity matrix S is defined as follows:

$$S_{ij} = \begin{cases} \exp\left(-\frac{1}{\delta}\left\|x_i - x_j\right\|^2\right) & \text{if } x_i \text{ is among k nearest neighbors of } x_j \\ & \text{or } x_j \text{ is among nearest neighbors of } x_i; \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

In many applications, such as face recognition, the dimensionality of the input data is typically larger than the number samples, i.e., $n \ll N$. The rank of XDX^T is at most n, while XDX^T is a $N \times N$ matrix. The matrix XDX^T is singular. LPP employs a procedure similar to Fisherface to overcome the singularity of XDX^T. The detailed algorithm of LPP is described as follows.

Step 1: PCA projection. Project data from the original space to PCA subspace vector with the projection matrix W_{PCA} by keeping 98 % information. For sake of simplicity, x_i is used to denote vector in the PCA subspace in the following steps.

Step 2: Construct the nearest neighbor graph and calculate the similarity matrix S. Let G be a graph with n nodes and ith node corresponding to x_i. An edge is put between nodes i and j, if x_i satisfies that $x_i$is among KNN of x_j, or $x_j$is among KNN of x_i and then calculate the similarity matrix S.

Step 3: Eigenmap. Calculate the projection matrix W_{LPP} and then calculate the projection matrix $W = W_{\mathrm{PCA}} W_{\mathrm{LPP}}$.

LPP is a very important approach to extract the feature vector with dimensionality reduction, but it suffering from small sample size (SSS) problem, i.e., XDX^T is singular. In order to overcome the singularity of XDX^T, LPP employs PCA to reduce the dimensionality. Firstly, the input vector is transformed to the low-dimensional PCA-transformed vector. Secondly, the nearest neighbor graph and the similarity matrix S are calculated with the PCA-transformed vectors with nearest neighbor criterion. Finally, the projection matrix is obtained through solving the eigenvalue problem.

We can analyze LPP with viewpoint. We compare other dimensionality reduction methods, PCA and LDA, with LPP. PCA is to seek only linear projection mapping to transform the original dimensional image space into a low-dimensional feature space. The optimal projection W_{PCA} of PCA is solved to maximize the determinant of the total scatter matrix of the transformed sample, i.e.,

$$W_{\mathrm{PCA}} = \arg\max_W \left|W^T S_T W\right| \tag{6.7}$$

where S_T is the total scatter matrix of the original sample vectors, i.e., $S_T = \sum_{i=1}^{n} (x_i - u)(x_i - u)^{\mathrm{T}}$. Different from the efficient directions of representation as PCA, LDA aims to seek the efficient direction of discrimination.

The optimal projection W_{LDA} is the matrix with orthonormal columns to maximize the ratio of the determinant of the between-class scatter matrix to the within-class scatter matrix in the projected space, which is defined as

$$W_{\mathrm{LDA}} = \arg\max_{W} \frac{|W^T S_B W|}{|W^T S_W W|} \tag{6.8}$$

where $S_B = \sum_{j=1}^{c} N_i (u_i - u)(u_i - u)^T$, $S_W = \sum_{i=1}^{c} \sum_{j=1}^{N_i} \left(x_i^j - u_i\right)\left(x_i^j - u_i\right)^T$, N_i is the number of samples in class i, and u_i is the mean vector of the class.

Both PCA and LPP are unsupervised learning methods, and LDA is supervised learning method. One of the differences between PCA and LPP lies in the global or local preserving property, that is, PCA seeks to preserve the global property, while LPP preserves the local structure. The locality preserving property leads to the fact that LPP outperforms PCA. Also as the global method, LDA utilizes the class information to enhance its discriminant ability which causes LDA to outperform PCA on classification. But the objective function of LPP is to minimize the local quantity, i.e., the local scatter of the projected data. This criterion cannot be guaranteed to yield a good projection for classification purposes. So it is reasonable to enhance LPP on classification using the class information like LDA.

6.3 Class-Wise Locality Preserving Projection

LPP is an unsupervised feature extraction without using the class label information to guide the training procedure. The label information can be used to guide the procedure of constructing the nearest neighbor graph for classification in many applications. In this section, we describe the CLPP algorithm in detail. CLPP uses the class information to model the manifold structure. One method is projecting two points belonging to the same class to the same point in the feature space. The similarity matrix S is defined as

$$S_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ belong to the same class;} \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

With the method shown in (6.9), CLPP is apt to over-fit the training data and sensitive to noise. We propose a novel class-wise way of constructing the nearest neighbor graph as follows.

$$S_{ij} = \begin{cases} \exp\left(-\frac{1}{\delta}\left\|x_i - x_j\right\|^2\right) & \text{if } x_i \text{ and } x_j \text{ belong to the same class;} \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

The Euclidean distance $\left\|x_i - x_j\right\|^2$ is in the exponent, and the parameter δ is used as a regulator. How to select the value of the parameter δ is still an open problem. We can control the overall scale or the smoothing of space through

changing the value of the parameter δ. For example, if x_i and x_j are not very close and the value of δ is very large, then the value of S_{ij} in Eq. (6.10) equals to 1. Moreover, if the value of δ is enough large, then the way of constructing the nearest neighbor graph shown in (6.9) is equivalent to the way (6.10). Say, the definition of S_{ij} in (6.10) is regarded as the generalization of that in (6.10). The way (6.9) uses the class label information to guide the training procedure, which improve the efficiency of training the model. But the method endures the free parameter selection problem. The performance of CLPP depends on whether the value of δ is appropriately chosen. An alternative way of constructing the nearest neighbor graph is proposed as follows:

$$S_{ij} = \begin{cases} \frac{x_i^T x_j}{\|x_i\| \|x_j\|} & \text{if } x_i \text{ and } x_j \text{ belong to the same class;} \\ 0 & \text{otherwise} \end{cases} \tag{6.11}$$

As shown in (6.11), the local structure and discriminative information of the data may be used for feature extraction. We have a comprehensive analysis on the above four ways as follows. The ways shown in (6.8), (6.9), (6.10), and (6.11) are denoted as S0, S1, S2, and S3, respectively. Suppose G be an adjacency graph with n nodes. S0 puts an edges between nodes i and j if x_i and x_j are "close." In other words, node i and node j are connected with one edge if x_i is among k nearest neighbors of x_j or x_j is among k nearest neighbors of x_i. S0 works well on the reconstruction, but it is not suitable to classification. For classification, the points in the feature space hopefully have the large class separability. S0 does not use the class label information to construct the nearest neighbor graph, so S0 is not so suitable to classification. S1 makes two points become the same point in the feature space if they belong to the same class, and the adjacency graph is corrupt because the nodes have centered into one point and the edges are disconnected. It contradicts the main idea of LPP. S2 considers the class label and local information together to construct the graph. Instead of putting an edge between nodes i and j if x_i and x_j are "close," S2 puts an edge between the nodes i and j if x_i and x_j belong to the same class, and it emphasizes the Euclidean distance $\left\|x_i - x_j\right\|^2$ of two points in the exponent. However, how to choose the value of parameter δ is still an open problem. S3 measures the similarity of two points with cosine similarity measure. CLPP takes advantage of local structure and class label information during the procedure of constructing the nearest neighbor graph, so CLPP expects to outperform LPP on feature extraction for classification.

6.4 Kernel Class-Wise Locality Preserving Projection

The nonlinear mapping Φ is used to map the input data $\mathbb{R}^N$ into a Hilbert space, i.e., $x \mapsto \Phi(x)$. We implement CLPP in the Hilbert space $\Phi(X) = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)]$. The objective function of CLPP in the Hilbert space is written as follows:

$$\min \sum_{i,j}^{n} \left\| z_i^{\Phi} - z_j^{\Phi} \right\|^2 S_{ij}^{\Phi} \tag{6.12}$$

where S_{ij}^{Φ} is a similarity matrix with weights characterizing the likelihood of two points in the Hilbert space and z_i^{Φ} is the one-dimensional representation of $\Phi(x_i)$ with a projection vector w^{Φ}, i.e., $z_i^{\Phi} = \left(w^{\Phi}\right)^T \Phi(x_i)$. The optimal transformation matrix w^{Φ} is calculated through solving an eigenvalue problem. Any eigenvector may be expressed by a linear combination of the observations in feature space as follows:

$$w^{\Phi} = \sum_{p=1}^{n} \alpha_p \Phi\left(x_p\right) = Q\alpha \tag{6.13}$$

where $Q = [\Phi(x_1)\, \Phi(x_2)\, \cdots\, \Phi(x_n)]$ and $\alpha = [\alpha_1\, \alpha_2 \cdots\, \alpha_n]$. We formulate CLPP with the dot product to generalize it to the nonlinear case. The dot product in the Hilbert space is presented with a kernel function, i.e., $k\left(x_i, x_j\right) = \Phi(x_i)^T \Phi\left(x_j\right)$. Accordingly, it is easy to generalize four ways, S0, S1, S2, S3, to the nonlinear cases. For example, S3 is extended to the following formulation with the kernel trick:

$$S_{ij}^{\Phi} = \begin{cases} \frac{k(x_i, x_j)}{\sqrt{k(x_i, x_i)}\sqrt{k(x_j, x_j)}} & \text{if } x_i \text{ and } x_j \text{ belong to the same class;} \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

Proposition 1 $\frac{1}{2}\sum_{i,j}^{n} \left\| z_i^{\Phi} - z_j^{\Phi} \right\|^2 S_{ij}^{\Phi} = \alpha^T K\left(D^{\Phi} - S^{\Phi}\right) K\alpha$, *where* S^{Φ} *is a similarity matrix calculated with Eq.* (6.13), *and* K *is kernel matrix calculated with the training samples, i.e.,* $K = Q^T Q$ *and* $D^{\Phi} = \operatorname{diag}\left[\sum_j S_{1j}^{\Phi}, \sum_j S_{2j}^{\Phi}, \ldots, \sum_j S_{nj}^{\Phi}\right]$.

Matrix D^{Φ} provides a natural measure on the data points. The bigger the value D_{ii}^{Φ} (corresponding to z_i^{Φ}) is, the more "important" is z_i^{Φ}. Therefore, we impose a constraint $\left(Z^{\Phi}\right)^T D^{\Phi} Z^{\Phi} = 1$, i.e., $\alpha^T K D^{\Phi} K\alpha = 1$. Then, minimization problem is transformed as

$$\min_{\alpha} \alpha^T K L^{\Phi} K\alpha$$

$$\text{Subject to } \alpha^T K D^{\Phi} K\alpha = 1 \tag{6.15}$$

where $L^{\Phi} = D^{\Phi} - S^{\Phi}$. Now, let us consider QR decomposition of matrix K, i.e., $K = P\Lambda P^T$, where $P = [r_1, r_2, \ldots, r_m]$, $(m \leq n)$ and $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$, and $r_1, r_2, \ldots, r_m$ are K's orthonormal eigenvector corresponding to m largest nonzero eigenvalue $\lambda_1, \lambda_2, \ldots, \lambda_m$. Let $\beta = \Lambda^{\frac{1}{2}} P^T \alpha$, i.e., $\alpha = P\Lambda^{-\frac{1}{2}}\beta$, we can reconsider Eq. (6.12) as follows:

$$w^{\Phi} = Q\alpha = QP\Lambda^{-\frac{1}{2}}\beta \tag{6.16}$$

Then, z_i^{Φ} is the one-dimensional representation of $\Phi(x_i)$ with a projection vector w^{Φ}, i.e.,.

$$z_i^{\Phi} = \beta^T \left(QP\Lambda^{-\frac{1}{2}}\right)^T \Phi(x_i) \tag{6.17}$$

Consequently, $z_i^{\Phi} = \beta^T y_i$, where

$$y_i = \left(QP\Lambda^{-\frac{1}{2}}\right)^T \Phi(x_i)$$

$$\begin{aligned} y_i &= \left(QP\Lambda^{-\frac{1}{2}}\right)^T \Phi(x_i) \\ &= \left(\frac{r_1}{\sqrt{\lambda_1}} \frac{r_2}{\sqrt{\lambda_2}} \cdots \frac{r_m}{\sqrt{\lambda_m}}\right)^T [k(x_1, x), k(x_2, x), \ldots, k(x_n, x)]^T \end{aligned} \tag{6.18}$$

where $r_1, r_2, \ldots, r_m$ are K's orthonormal eigenvector corresponding to m largest nonzero eigenvalue $\lambda_1, \lambda_2, \ldots, \lambda_m$. The transformation in (6.18) is exactly KPCA transformation. We reconsider the objective function (6.11) in the KPCA-transformed space:

$$\min \sum_{i,j}^{n} \left\|\beta^T y_i - \beta^T y_j\right\|^2 S_{ij}^{\Phi} \tag{6.19}$$

where y_i be the KPCA-transformed feature vector. Then, we use the same method of solving the minimization problem in (6.1) to solve the Eq. (6.19). $YD^{\Phi}Y^T (Y = [y_1, y_2, \ldots, y_n])$ is a $m \times m$ matrix, $m \leq n$, so $YD^{\Phi}Y^T$ is nonsingular. The procedure of KCLPP is divided into three steps:

1. Project x_i to y_i with the KPCA transformation by keeping the 98 % information.
2. Construct the similarity matrix S^{Φ} with kernel function.
3. Project y_i to the KCLPP-transformed feature z_i^{Φ} with $z_i^{\Phi} = \left(W_{\text{KCLPP}}^{\Phi}\right)^T y_i$, where W_{KCLPP}^{Φ} is a matrix whose column vectors are the d eigenvectors corresponding to the first d smallest eigenvalues.

Given a training set consisting of N-dimensional data $\{x_1, x_2, \ldots, x_n\}$ in $\mathbb{R}^N$ from the L classes, we firstly train CLPP (or KCLPP) as follows.

Step 1. Project N-dimensional data $\{x_1, x_2, \ldots, x_n\}$ to a lower-dimensional vector $\{y_1, y_2, \ldots, y_n\}$ through PCA (KPCA if KCLPP) transformation by keeping the 98 % information.

Step 2. Construct the nearest neighbor graph with the selected method. If CLPP, construct the nearest neighbor graph of the data $\{y_1, y_2, \ldots, y_n\}$ with S1 (or S2, S3); if KCLPP, construct the similarity matrix S^{Φ} with $\{x_1, x_2, \ldots, x_n\}$with kernel version of S1 (or S2, S3).

Step 3. Train the projection matrix W_{CLPP} ($W^{\Phi}_{\mathrm{KCLPP}}$ if KCLPP), where W_{CLPP} ($W^{\Phi}_{\mathrm{KCLPP}}$ if KCLPP) is a matrix whose column vectors are the d eigenvectors corresponding to the first d smallest eigenvalues.

Step 4. Extract the feature $z_i\left(z_i^{\Phi}\text{ if KCLPP}\right)$ with $z_i = (W_{\mathrm{CLPP}})^T y_i$ ($z_i^{\Phi} = \left(W^{\Phi}_{\mathrm{KCLPP}}\right)^T y_i$ if KCLPP).

After feature extraction, the nearest neighbor (to the mean) classifier with the similarity measure δ is applied to classification:

$$\delta\left(F, M_k^0\right) = \min_j \quad \delta\left(F, M_j^0\right) \tag{6.20}$$

where M_k^0, $k = 1, 2, \ldots, L$, is the mean of the training samples for class ω_k. The feature vector F is classified as belonging to the class of the closet mean, M_k^0, using the similarity measure δ. Another classifier, such as Fisher classifier, is also used to classification, and we evaluate their performance in the experiments.

6.5 Kernel Self-Optimized Locality Preserving Discriminant Analysis

In this section, firstly we improve the LPP with class labels and propose the nonparametric similarity measure to construct the nearest neighbor graph, and secondly, we extend the nonparametric similarity with data-dependent kernel to propose a novel constraint optimization equation for KSLPDA, and finally, the detail procedure of solving the constraint optimization equation for KSLPDA is presented.

6.5.1 Outline of KSLPDA

Firstly, we extend LPP with the corresponding kernel nonparametric similarity measure, and secondly, we use the data-dependent kernel to kernel LPP with kernel self-optimization to present kernel self-optimization locality preserving discriminant analysis (KSLPDA).

The main idea of kernel method is to map data into a feature space with a nonlinear mapping where the inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly. We apply the nonlinear mapping to embed the nonlinear mapping Φ to map the input data space $\mathbb{R}^N$ into the feature space F:$\Phi : \mathbb{R}^N \rightarrow F$ and $x \mapsto \Phi(x)$. Correspondingly, a pattern in the original input space $\mathbb{R}^N$ is mapped into a potentially much higher-dimensional feature vector in the feature space F with $z^{\Phi} = \left(w^{\Phi}\right)^T \Phi(x)$. Based on

LPP, kernel LPP aims to seek the data points $\{z_1, z_2, \ldots, z_n\}$ with the same locality neighborhood structure as $\{\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)\}$ in the nonlinear mapping space. Then, the objective function is

$$\min \sum_{i,j}^{n} \left\| z_i^{\Phi} - z_j^{\Phi} \right\|^2 S_{ij}^{\Phi} \tag{6.21}$$

Accordingly, the transformation matrix w^{Φ} is achieved through solving a generalized eigenvalue problem to minimize the objective function. So $w^{\Phi} = \sum_{p=1}^{n} \alpha_p \Phi(x_p) = Q\alpha$, where $Q = [\Phi(x_1)\, \Phi(x_2) \cdots \Phi(x_n)]$ and $\alpha = [\alpha_1\, \alpha_2 \cdots \alpha_n]$. So it is easy to obtain the kernel matrix $K_0 = Q^T Q$. Three popular kernel functions are polynomial kernels $k_0(x, y) = (x \cdot y)^d$, $d \in N$, Gaussian kernels $k_0(x, y) = \exp\left(-(1/2\sigma^2)\|x - y\|^2\right)$, $\sigma > 0$, and sigmoid kernels $k_0(x, y) = \tanh(k(x \cdot y) + \upsilon)$, $(k > 0, \upsilon < 0)$.

In many applications, the parameters of kernel function are chosen from discrete values. But the geometrical structure of kernel is not changing with the change in kernel parameters. The inappropriate choosing of kernel function will influence the performance of the kernel LPP. We introduce a novel kernel, namely the data-dependent kernel to improve the kernel LPP. Data-dependent kernel with a general geometrical structure can obtain the different kernel structure with different combination parameters, and the parameters are self-optimized under the criterions. The data-dependent kernel $k(x, y) = f(x)f(y)k_0(x, y)$, where $k_0(x, y)$ is a basic kernel, i.e., polynomial kernel, Gaussian kernel, and so on, and $f(x)$ is a positive real-valued function of x. In the previous work [34], researchers used $f(x) = \sum_{i \in SV} a_i e^{-\delta \left\| x - \widetilde{x}_i \right\|^2}$, where $\widetilde{x}_i$ is the ith support vector, SV denotes support vectors, a_i is a positive number representing the contribution of $\widetilde{x}_i$, and δ is a free parameter. We generalize Amari and Wu's method as $f(x) = b_0 + \sum_{n=1}^{N_{XV}} b_n e(x, \tilde{x}_n)$ in our previous work [33, 35], where δ is the free parameter, $\tilde{x}_i$ is the expansion vectors (XVs), N_{XV} is the number of expansion vectors, and b_n ($n = 0,1,2,\ldots,N_{\mathrm{XVs}}$) are the corresponding expansion coefficients. Supposed that $D = \mathrm{diag}(f(x_1), f(x_2), \ldots, f(x_n))$, the relation between the data-dependent kernel matrix K and the basic kernel K_0 is described as $K = DK_0D$ and then $D1_n = E\beta$ where $\beta = [b_0, b_1, b_2, \ldots, b_{N_{\mathrm{XVs}}}]^T$ and the matrix $E = [1e(x_i, \widetilde{x}_1) \cdots e(x_i, \widetilde{x}_{N_{\mathrm{XVs}}})]_{n \times N_{\mathrm{XVs}}}$. Given the basic kernel matrix K_0 and the matrix E, the data-dependent kernel matrix is

$$K^{(\beta)} = f(K_0, E, \beta) \tag{6.22}$$

where $f(\cdot)$ is determined with $K = DK_0D$ and $D1_n = E\beta$. The kernel structure is changeable with changing β. Based on LPP, it is easy to achieve $\frac{1}{2}\sum_{i,j}^{n} \left\| z_i^{\Phi} - z_j^{\Phi} \right\|^2 S_{ij}^{\Phi} = \alpha^T K^{(\beta)} \left(D^{\Phi} - S^{\Phi}\right) K^{(\beta)} \alpha$, where S^{Φ} is a similarity matrix with

samples and $D^{\Phi} = \text{diag}\left[\sum_j S_{1j}^{\Phi}, \sum_j S_{2j}^{\Phi}, \ldots, \sum_j S_{nj}^{\Phi}\right]$. According to the definition of C3, the data-dependent kernel nonparametric similarity measure is defined as CS3:

$$S_{ij} = \begin{cases} \frac{k(x_i, x_j)}{\sqrt{k(x_i, x_i)}\sqrt{k(x_j, x_j)}} & \text{if } x_i \text{ and } x_j \text{ belong to the same class;} \\ 0 & \text{otherwise} \end{cases} \tag{6.23}$$

The class-wise similarity measures with kernels have the same functions for the class-wise similarity measures. Matrix D^{Φ} provides a natural measure on the data points. The bigger the value D_{ii}^{Φ} (corresponding to z_i^{Φ}) is, the more "important" is z_i^{Φ}. Consider the constraint $(Z^{\Phi})^T D^{\Phi} Z^{\Phi} = 1$, that is, $\alpha^T K^{(\beta)} D^{\Phi} K^{(\beta)} \alpha = 1$. Since $L^{\Phi} = D^{\Phi} - S^{\Phi}$, an optimization problem can be obtained as

$$\min_{\alpha,\beta} \alpha^T K^{(\beta)} L^{\Phi} K^{(\beta)} \alpha$$

$$\text{Subject to } \alpha^T K D^{\Phi} K \alpha = 1 \text{ and } \beta^T \beta = 1 \tag{6.24}$$

KSLPDA is to solve (6.24) to obtain the optimal α and β. β is to optimize the data-dependent kernel structure, and α is to construct the project matrix for locality preserving projection. So the procedure of the KSLPDA is divided into two main steps. One is to construct the optimal kernel structure through solving the optimal parameter β^*, and the second is to seek the optimal projection matrix of α with β^*.

Step 1 **Solving β**

Now, our goal is to seek the optimal parameter β^* through solving the constraint optimal equation. Since KSLPDA is used in classification task, the data from the input space into the feature space have the largest discriminative ability with the optimal kernel structure. We use the Fisher criterion and maximum margin criterion to construct the constraint optimization equation to solve β, and the class separability of data is achieved for classification with this optimal parameter β^*.

Firstly, we construct the constraint optimization equation with Fisher criterion. The class discriminative ability of the data in empirical feature space is defined as $J_{\text{Fisher}} = \text{tr}(S_B^{\Phi}) / \text{tr}(S_W^{\Phi})$, where J_{Fisher} measures the separable scalar, S_B^{Φ} is between-class scatter matrix, S_W^{Φ} is the within-class matrix, and tr is the trace of matrix. Given $B = \text{diag}\left(\frac{1}{n_1}Kc_{11}, \frac{1}{n_2}Kc_{22}, \ldots, \frac{1}{n_L}Kc_{LL}\right) - \frac{1}{n}K_{\text{total}}$ and $W = \text{diag}(k_{11}, k_{22}, \ldots, k_{nn}) - \text{diag}\left(\frac{1}{n_1}Kc_{11}, \frac{1}{n_2}Kc_{22}, \ldots, \frac{1}{n_L}Kc_{LL}\right)$, where $\text{tr}(S_B^{\Phi}) = 1_n^T B 1_n$ and $\text{tr}\left(S_W^{\Phi}\right) = 1_n^T W 1_n$, where $Kc_{ij}(i, j = 1, 2, \ldots, L)$ is a data-dependent kernel matrix calculated with the ith and jth class samples and the data-dependent kernel matrix K_{total} with its elements k_{ij} calculated with pth and qth samples. Since $K = DK_0D$, $D1_n = E\beta$, $B = DB_0D$, and $W = DW_0D$, and $\beta^T\beta = 1$, the constraint optimization equation is defined as

$$\max \quad J_{\text{Fisher}}(\beta)$$
$$\text{subject} \quad \text{to} \quad \beta^T\beta - 1 = 0 \tag{6.25}$$

where $J_{\text{Fisher}}(\beta) = \left(\beta^T E^T B_0 E\beta\right)/\left(\beta^T E^T W_0 E\beta\right)$, $E^T B_0 E$ and $E^T W_0 E$ are constant matrices. Many algorithms are proposed for the solution of the above optimization equation [36]. The iteration algorithm is shown as

$$\frac{\partial J_{\text{Fisher}}(\beta)}{\partial\beta} = \frac{2}{J_2^2}\left(J_2 E^T B_0 E - J_1 E^T W_0 E\right)\beta \tag{6.26}$$

Let $\dfrac{\partial J_{\text{Fisher}}(\beta)}{\partial\beta} = 0$, then $J_1 E^T W_0 E\beta = J_2 E^T B_0 E\beta$. If $\left(E^T W_0 E\right)^{-1}$ exists, then

$$J_{\text{Fisher}}\beta = \left(E^T W_0 E\right)^{-1}\left(E^T B_0 E\right)\beta \tag{6.27}$$

The optimal expansion vector β is obtained through solving the eigenvalue of $\left(E^T W_0 E\right)^{-1}\left(E^T B_0 E\right)$. But in many practical applications, $\left(E^T W_0 E\right)^{-1}\left(E^T B_0 E\right)$ is not symmetric or $E^T W_0 E$ is singular. The optimal solution β^* is solved through the iteration algorithm [36], then

$$\beta^{(n+1)} = \beta^{(n)} + \varepsilon\left(\frac{1}{J_2}E^T B_0 E - \frac{J_{\text{Fisher}}}{J_2}E^T W_0 E\right)\beta^{(n)} \tag{6.28}$$

ε is the learning rate, then $\varepsilon(n) = \varepsilon_0(1 - n/N)$, where ε_0 is the initial learning rate, and n and N are the iteration number and total iteration number.

Secondly, we create the constraint optimization equation with maximum margin criterion to solve β. The average margin $J_{\text{mmc}} = \frac{1}{2n}\sum_{i=1}^{L}\sum_{j=1}^{L} n_i n_j d\left(c_i, c_j\right)$ between classes c_i and c_j, where $d\left(c_i, c_j\right) = d\left(m_i^\Phi, m_j^\Phi\right) - S(c_i) - S(c_j)$ denotes the margin between any two classes, and $S(c_i)$ denotes the measure of the scatter of class c_i and $d\left(m_i^\Phi, m_j^\Phi\right)$ denotes the distance between the means of two classes. Supposed that $\text{tr}(S_i^\Phi)$ measures the scatter of the class i, then it is easy to obtain $\text{Dis} = \text{tr}\left(2S_B^\Phi - S_T^\Phi\right)$. Since $Y = KP\Lambda^{-1/2}$ and $K = P\Lambda^T P^T$ in the kernel empirical feature space, $Y_0 = K_0 P_0 \Lambda_0^{-1/2}$ and $K_0 = P_0\Lambda_0^T P_0^T$, it is easy to achieve $\text{trace}(S_T) = \beta^T(X_T)^T X_T\beta$, $\text{trace}(S_B) = \beta^T X_B^T X_B\beta$, where $X_T = \left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)E$, $X_B = Y_0 M^T E$, $M = M_1 - M_2$, $M_1 = \text{diag}\left\{\left[\frac{1}{\sqrt{m_i}}\right]_{m_i\times m_i}\right\}, (i = 1, 2, \ldots, c)$, $M_2 = \text{diag}\left\{\frac{1}{m}\sum_j^c \sqrt{m_j}\right\}$. Then, the optimization equation is

$$\max_{\beta} \quad J_{\text{mmc}}(\beta)$$

$$\text{subject} \quad \text{to} \quad \beta^T\beta - 1 = 0 \tag{6.29}$$

where $J_{\text{mmc}}(\beta) = \text{trace}\big(\beta^T(2\tilde{S}_B - \tilde{S}_T)\beta\big)$, $\tilde{S}_B = X_B X_B^T$ and $\tilde{S}_T = X_T X_T^T$. The optimal β^* equals to the eigenvector of $2\tilde{S}_B - \tilde{S}_T$ corresponding to the largest eigenvalue. Due to $P^T\tilde{S}_B P = \Lambda$ and $P^T\tilde{S}_T P = I$, where $P = \phi\theta^{-1/2}\psi$, θ and ϕ are the eigenvalue and eigenvector of $\tilde{S}_T$, ψ is the feature matrix of $\theta^{-1/2}\phi^T\tilde{S}_B\phi\theta^{-1/2}$, then P is the eigenvector of $2\tilde{S}_B - \tilde{S}_T$ corresponding to $2\Lambda - I$.

Thirdly, we present four versions of $e(x, \tilde{x}_n)$ marked by E1, E2, E3, and E4 [33], shown as follows:

E1: $e(x, \tilde{x}_n) = \begin{cases} 1 & x \text{ and } x_n \text{ come from the same class} \\ e^{-\delta\|x - x_n\|^2} & x \text{ and } x_n \text{ come from the different class} \end{cases}$, which regards all the training samples with the class labels as the expansion vectors, and considers the class information of samples and expects that the samples from the same classes will centralize into one point.

E2: $e(x, \tilde{x}_n) = e^{-\delta\|x - x_n\|^2}$, $(n = 1, 2, \ldots, M)$, which expects all training samples as the expansion vectors, and is equivalent to the traditional method. In many practical applications, such as linear discriminant analysis, all training samples are considered as support vectors. In this case, the number of training samples is equal to the number of expansion vectors.

E3: $e(x, \tilde{x}_n) = \begin{cases} 1 & x \text{ and } \bar{x}_n \text{ come from the same class} \\ e^{-\delta\|x - \bar{x}_n\|^2} & x \text{ and } \bar{x}_n \text{ come from the different class} \end{cases}$, where $\bar{x}_n$ is the mean vector of all samples, $N_{\text{XV}} = L$. This method is proposed for the computation of E1 and E2, the above two methods regard all training samples as expansion vectors. Not so heavy computation stress incurs for the small number of training samples, but it will bring big computation stress for the large number of training samples. This method regards the mean vectors of all samples as expansion vectors; this method considers the distribution of center of all training samples, and the class labels of all training samples are considered.

E4: $e(x, \tilde{x}_n) = e^{-\delta\|x - \bar{x}_n\|^2}$, $(n = 1, 2, \ldots, L)$, which regards the mean vector of samples as the expansion vector. This method only considers the distance between any sample and the center of all samples without considering the class label of each sample.

Step 2 **Solving** α

After the optimal β^* is obtained, we seek the optimal projection α as follows:

$$\min_{\alpha} \alpha^T K^{(\beta^*)} L^{\Phi} K^{(\beta^*)} \alpha$$

$$\text{Subject to } \alpha^T K^{(\beta^*)} D^{\Phi} K^{(\beta^*)} \alpha = 1 \tag{6.30}$$

Supposed that $r_1, r_2, \ldots, r_m$ are K's orthonormal eigenvector corresponding to m largest nonzero eigenvalue $\lambda_1, \lambda_2, \ldots, \lambda_m$. i.e., $K = P\Lambda P^T$ with QR decomposition, where $P = [r_1, r_2, \ldots, r_m]$ and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Let $\varphi = \Lambda^{\frac{1}{2}} P^T \alpha$, i.e., $\alpha = P\Lambda^{-\frac{1}{2}}\varphi$, $L_k = \Lambda^{\frac{1}{2}} P^T L^{\Phi} P \Lambda^{\frac{1}{2}}$, $D_k = \Lambda^{\frac{1}{2}} P^T D^{\Phi} P \Lambda^{\frac{1}{2}}$, then

$$L(\varphi, \lambda) = \varphi^T L_k \varphi - \lambda\left(\varphi^T D_k \varphi - I\right) \tag{6.31}$$

with the parameter λ. The Lagrangian L must be minimized with respect to λ and φ. The eigenvalue decomposition is used to solve the above optimization equation. Let $\lambda^* = 1/\lambda$, then we can solve the eigenvectors of the generalized equation problem shown as follows:

$$D_k \varphi = \lambda^* L_k \varphi \tag{6.32}$$

The solution of the above constrained optimization problem is equal to the eigenvector corresponding to the largest eigenvalue. The detailed procedure of solving the Eq. (6.22) is presented in the previous work [24].

Step 3 **Classification with Fisher classifier**

After the features are extracted with KSLPDA, then the dimensionality of the KSLPDA-based feature is reduced with Fisher discriminant analysis [4] to the $c-1$ of dimensionality, where c denotes the class number of samples, and finally, we apply the nearest neighbor (to the mean) rule for classification with some similarity (distance) measure.

Supposed that y is the n-dimensional feature vector extracted by KSLPDA, the classification procedure with Fisher classifier is described as follows.

Firstly, the feature vector y is reduced into $c-1$-dimensional vector z for classification with Fisher discriminant analysis. We calculate the optimal Fisher projection matrix W_F to maximize the Fisher ratio, that is, $W_F = \arg\max_W \{|W^T S_b W|/|W^T S_w W|\}$, where $S_B = \sum_{j=1}^{c} N_i (u_i - u)(u_i - u)^T$, $S_W = \sum_{i=1}^{c} \sum_{j=1}^{N_i} \left(y_i^j - u_i\right)\left(y_i^j - u_i\right)^T$, N_i is the number of samples in class i and u_i is the mean vector of the class. Then, the $c-1$-dimensional vector z is calculated with $z = w_F^T y$.

Secondly, we apply the nearest neighbor (to the mean) rule to classify the $c-1$-dimensional vector z. Supposed that C_k be the mean of the training samples for class ω_k, the $c-1$-dimensional vector z is classified into the class using the rule $\Gamma(z, C_k) = \min_j \Gamma\left(z, C_j\right) \rightarrow z \in \omega_k$ based on the similarity measure Γ and the Euclidean distance measure δ_L.

Table 6.1 Deterministic training and test set on ORL dataset

	ORL_A	ORL_B	ORL_C	ORL_D	ORL_E
Training set	1#, 2#, 3#, 4#, 5#	10#, 1#, 2#, 3#, 4#	9#, 10#, 1#, 2#, 3#	8#, 9#, 10#, 1#, 2#	7#, 8#, 9#, 10#, 1#,
Testing set	6#, 7#, 8#, 9#, 10#	5#, 6#, 7# ,8# ,9#	4#, 5#, 6#, 7#, 8#	3#, 4#, 5#, 6#, 7#	2#, 3#, 4#, 5#, 6#

Table 6.2 Deterministic training and test set on YALE dataset

	YALE_a	YALE_b	YALE_c	YALE_d	YALE_e
Training set	1#, 2#, 3#, 4#, 5#	11#, 1#, 2#, 3#, 4#	9#, 10#, 11#, 1#, 2#,	8#, 9#, 10#, 11#, 1#	7#, 8#, 9#, 10#, 11#
Testing set	6#, 7#, 8#, 9#, 10#, 11#	5#, 6#, 7#, 8#, 9#, 10#	3#, 4#, 5#, 6#, 7#, 8#	2#, 3#, 4#, 5#, 6#, 7#	1#, 2#, 3#, 4#, 5#, 6#

Notes 1# denotes the first image of each person, 2# denotes the second image of each person, and other images are marked with the same ways

6.6 Experiments and Discussion

6.6.1 Experimental Setting

ORL face database, developed at the Olivetti Research Laboratory, Cambridge, UK, is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations in the images are across pose, time, and facial expression. YALE face database was constructed at the YALE Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting conditions (left-light, center-light, and right-light), and different facial expressions (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. To reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels. Similarly, the images from YALE databases are cropped to the size of 100×100 pixels.

We implement experiments with two manners. (1) A deterministic manner. The training set and the testing set are constructed as shown in Tables 6.1 and 6.2. Here, our goal is to have a good look at the performance of specific partition of the database; thus, we can see how much the influence of recognition rate under the different PIE (pose, illumination, and expression). (2) A random manner. From ORL face database, we randomly select 5 images from each subject, 200 images in total for training, and the rest 200 images are used to test the performance. Five images of each person randomly selected from YALE database are used to construct the training set, and the rest 5 images of each person are used to test the performance of the algorithms.

Table 6.3 Recognition accuracy under different k and δ for S0 on ORL face database

	$\delta = 10^4$	$\delta = 10^5$	$\delta = 10^6$	$\delta = 10^7$	$\delta = 10^8$
$k = 1$	0.6100	0.6450	0.8300	0.7635	0.7350
$k = 2$	0.6000	0.7950	0.8500	0.7900	0.7800
$k = 3$	0.6650	0.8350	0.8000	0.7600	0.7500
$k = 4$	0.6650	0.9050	0.7700	0.7500	0.7400
$k = 5$	0.6650	0.8700	0.7500	0.6735	0.6950

Table 6.4 Recognition accuracy under different k and δ for S0 on YALE face database

	$\delta = 10^5$	$\delta = 10^6$	$\delta = 10^7$	$\delta = 10^8$	$\delta = 10^9$	$\delta = 10^{10}$
$k = 1$	0.6222	0.4589	0.8111	0.7856	0.8133	0.8000
$k = 2$	0.6111	0.5778	0.7222	0.7111	0.7000	0.7000
$k = 3$	0.7000	0.6111	0.6756	0.6222	0.6111	0.5778
$k = 4$	0.7000	0.6000	0.7000	0.6778	0.6556	0.6667
$k = 5$	0.7000	0.6333	0.6833	0.7000	0.7111	0.7111

6.6.2 Procedural Parameters

We choose the procedural parameters for each algorithm with the cross-validation method. These procedural parameters are summarized as follows. (1) k and δ of S0 for LPP; (2) the best value of the δ of S1 for CLPP1; (3) the kernel parameters for KCLPP and KPCA. Moreover, the dimensionality of feature vector is set to 60 on ORL database, and the dimensionality of feature vector is set to 40 on YALE database.

As shown in Tables 6.3 and 6.4, the parameters $k = 4$ and $\delta = 10^5$ are chosen on ORL dataset, and the parameters $k = 1$ and $\delta = 10^7$ are chosen on YALE

Table 6.5 Recognition accuracy under δ for S2 on ORL face database

δ	10^4	10^5	10^6	10^7
Recognition rate (%)	61.20	84.40	93.60	93.80

Table 6.6 Recognition accuracy under δ for S2 on YALE face database

δ	10^6	10^7	10^8	10^9
Recognition rate (%)	53.36	86.22	95.11	95.33

Table 6.7 Recognition accuracy (%) under four linear methods of constructing the nearest neighbor graph

	S0	S1	S2	S3
ORL	87.00	92.00	94.50	95.00
YALE	70.58	91.33	94.44	95.11

Table 6.8 Selection of kernel parameters on ORL dataset

Kernels	Polynomial kernel			Gaussian kernel			
	$d=1$	$d=2$	$d=3$	$\sigma^2=1\times10^7$	$\sigma^2=1\times10^8$	$\sigma^2=1\times10^9$	$\sigma^2=1\times10^{10}$
Recognition rate (%)	95.00	95.50	94.50	91.00	94.00	95.00	94.00

database. And the procedural parameter of S2 is $\delta = 10^7$ on ORL database and $\delta = 10^9$ on YALE database, which are shown in Tables 6.5 and 6.6.

We compare four ways of constructing the nearest neighbor graph on the recognition performance. As shown in Table 6.7, S3 outperforms other three ways on recognition performance. In the next experiments, S3 is chosen to construct the nearest neighbor graph.

Polynomial kernel $k(x,y) = (x \cdot y)^d \ \ (d \in N)$ and Gaussian kernel $k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$ $(\sigma > 0)$ with the best kernel parameters are chosen in the following experiments. As shown in Table 6.8, KCLPP achieves the highest recognition accuracy on ORL dataset where polynomial kernel with $d = 2$, while Gaussian kernel with $\sigma^2 = 1 \times 10^9$, is chosen for KCLPP on YALE database.

6.6.3 Performance Evaluation of KCLPP

In this section, we evaluate the proposed algorithm on computation efficiency and recognition accuracy. The time consumption of calculating the projection matrix is used to evaluate the algorithms on computation efficiency [17]. We compare PCA, KPCA, LPP, CLPP, and KCLPP on computation efficiency. The procedure of CLPP and LPP is divided into three steps: PCA projection, constructing the nearest neighbor graph, and eigenmap. KCLPP is implemented with three steps: KPCA projection, constructing the nearest neighbor graph, and eigenmap. Accordingly, we calculate the time cost of executing KCLPP with sum of time cost of above three steps. The results on two datasets are shown in Table 6.9. Firstly, the linear methods cost less time compared with their corresponding kernel versions owing to the high time consumption of calculating the kernel matrix. Secondly, both LPP and CLPP achieve the lower computation efficiency than PCA because PCA is the one step of the procedure of LPP and CLPP. Finally, CLPP costs less time than LPP because CLPP constructs the nearest neighbor graph under the guidance of

Table 6.9 Computation cost (seconds) in calculating the projection matrices

	PCA	KPCA	LPP	CLPP	KCLPP
ORL	0.9409	5.7252	1.6448	1.3217	5.7897
YALE	0.8705	2.3794	1.2709	1.2133	2.4539

Table 6.10 Recognition accuracy (%) on YALE dataset with a deterministic manner

	PCA+LDA	LPP+LDA	CLPP+LDA	KPCA+LDA	KCLPP+NNC	KCLPP+LDA
YALE_a	92.00	86.33	90.00	94.44	94.44	86.67
YALE_b	92.00	90.67	91.11	92.22	92.22	86.67
YALE_c	93.00	88.56	86.67	93.33	93.33	74.44
YALE_d	92.00	88.89	90.00	93.33	93.33	90.00
YALE_e	90.50	95.56	93.33	96.67	96.67	80.00

Table 6.11 Recognition accuracy (%) on ORL dataset with a deterministic manner

	PCA+LDA	LPP+LDA	CLPP+LDA	KPCA+LDA	KCLPP+NNC	KCLPP+LDA
ORL_A	92.00	95.00	96.00	93.50	95.50	96.50
ORL_B	92.00	93.50	94.00	93.50	95.50	95.50
ORL_C	93.00	95.50	97.00	94.00	97.00	98.50
ORL_D	92.00	93.50	94.50	93.50	95.50	96.00
ORL_E	90.50	91.50	92.50	91.00	92.00	96.00

class label information and need not search the whole training samples set for seek the k nearest neighbors as LPP.

After evaluating the performance on computation efficiency, we test the recognition performance of the proposed algorithm. On the two databases with the deterministic, we implement the following algorithms: KCLPP+nearest neighbor (to mean) classifier, CLPP+LDA, LPP+LDA, KCLPP+LDA, PCA+LDA, KPCA+LDA, PCA+NNC, KPCA+NNC. All above algorithms are implemented with the best procedural parameters. As results shown in Tables 6.10 and 6.11, CLPP performs better than LPP when the same classifier is chosen for classification. KCLPP achieves the higher recognition accuracy than CLPP. The results show that LPP is improved with the class label information. Kernel method improves the classification performance of linear feature extraction methods. For example, KPCA and KCLPP outperform PCA and CLPP, respectively. LPP, CLPP, and its kernel version outperform PCA and its kernel version, which demonstrates the advantage of the local structure-based feature extraction methods.

Experiments on two databases have been systematically implemented, and the experiments reveal a number of interesting points which are summarized as follows: (1) The procedural parameters have heavy impact on performance of the algorithm, and the algorithm achieves the different performance with the different parameters. For example, two procedural parameters are chosen to construct the nearest neighbor graph for LPP, while CLPP is successful to avoid the parameter selection problem owing to the nonparametric way of constructing the nearest neighbor graph. Thereof, CLPP hopefully achieves the consistent recognition performance without influencing the parameter selection. (2) On the computation efficiency, CLPP also outperforms LPP because LPP does k nearest neighbor search, while CLPP uses the class label information to guide the procedure of constructing the nearest neighbor graph without searching in the whole sample set.

Table 6.12 MMC versus Fisher with four methods of choosing XVs on ORL face database

		E1	E2	E3	E4
Recognition accuracy	MMC	0.9365	0.9340	0.9335	0.9355
	FC	0.9245	0.9215	0.9210	0.9240
Time consumption (second)	MMC	0.05	0.05	0.02	0.02
	FC	1.50	1.48	0.20	0.19

Table 6.13 MMC versus Fisher with four methods of choosing XVs on YALE face database

		E1	E2	E3	E4
Recognition accuracy	MMC	0.9187	0.9227	0.9200	0.9227
	FC	0.9000	0.9147	0.9079	0.9187
Time consumption (seconds)	MMC	0.03	0.05	0.02	0.02
	FC	0.32	0.32	0.08	0.06

Table 6.14 Kernel self-optimization performance on ORL and YALE databases

	ORL	YALE
Kernel self-optimization-based KDA	0.9410	0.9267
KDA without kernel optimization	0.9250	0.9040

6.6.4 Performance Evaluation of KSLPDA

We randomly select 5 images from each subject from ORL face database for training, and the rest of the images are used to test the performance. Similarly, 5 images of each person randomly selected from YALE database are used to construct the training set, and the rest of the images of each person are used to test the performance of the algorithm. Fisher classifier is applied for classification, and the average recognition accuracy is used to evaluate kernel self-optimization. Four methods of choosing expansion vector and two kernel self-optimization methods are evaluated on ORL and YALE databases. As shown in Tables 6.12 and 6.13, MMC achieves a higher recognition accuracy and computation efficiency compared with FC. The computation efficiency of FC and MMC is evaluated through comparing the time consumption of solving the expansion coefficient vector between FC and MMC. The four methods achieve approximately the same recognition accuracy but the different computation efficiency. E3 and E4 outperform E1 and E2 on the computation efficiency. MMC-based kernel self-optimization method is applied to solve the expansion coefficients. The performance of kernel self-optimization is shown in Table 6.14.

In this section, we evaluate the performance of the proposed algorithms with the ORL, YALE, Wisconsin Diagnostic Breast Cancer (WDBC), and UCI databases. We evaluate the computation efficiency on ORL and YALE databases and compare the recognition accuracy on ORL, YALE, WDBC, and UCI databases. The optimal procedural parameters of each algorithm were chosen through the cross-

Table 6.15 Computation cost (seconds) in calculating the projection matrices

	PCA	KDA	LPP	CLPP	KCLPP	KSLPDA
ORL	0.9409	5.7252	1.6448	1.3217	5.7897	8.7567
YALE	0.8705	2.3794	1.2709	1.2133	2.4539	4.8693

validation method. We evaluate class-wise LPP (CLPP) and kernel class-wise LPP (KCLPP) proposed in the previous work [24]. CLPP and KCLPP are the representative algorithms of supervised LPP and kernel supervised LPP, respectively. Moreover, we also implement the traditional LPP [13] together with PCA [4] and KDA [37]. We compare their recognition performance on the computation efficiency and recognition accuracy. For the computation efficiency, we apply the time consumption of calculating the projection matrix to evaluate the algorithms on computation efficiency with random manner. With the random manner, 5 samples are randomly selected as the training samples from ORL database with the rest samples as the test samples. Similarly, 5 samples are chosen as the training dataset from YALE face database, and the rest face images are selected as the test dataset. For the recognition accuracy, we evaluate the algorithms with the determined manner. With the determined manner, we denote the subdataset with ORL_SD1, ORL_SD2,…, ORL_SD5 on ORL database, and YALE_SD1, YALE_SD2,…, YALE_SD5 on YALE database with the goal of analyzing the influence of PIE (pose, illumination, and expression) conditions on recognition accuracy. We construct the subdataset with 5 training samples and 5 test samples in ORL face database, and 5 training samples and 6 test samples per class are chosen to construct the subdataset of YALE database. Firstly, we evaluate the computation efficiency on the ORL and YALE databases with the random manner. As shown in Table 6.15, the linear methods cost less time compared with their corresponding kernel versions owing to the high time consumption of calculating the kernel matrix. Both LPP and CLPP achieve the lower computation efficiency than PCA because PCA is one step of the procedure of LPP and CLPP. CLPP costs less time than LPP because the nearest neighbor graph in CLPP is created with the guidance of class label information but without searching the whole training sample set for k nearest neighbors. But the KSLPDA costs more time because of the time cost of computing the optimal procedural parameter of the data-dependent kernel for the high recognition accuracy. The procedure of CLPP and LPP is divided into three steps: PCA projection, constructing the nearest neighbor graph, and eigenmap. KCLPP is divided into kernel mapping and CLPP, and the proposed KSLPDA is divided into kernel self-optimization and KCLPP. KSLPDA has the highest time consumption.

Secondly, we evaluate the algorithms on the recognition accuracy with ORL and YALE databases. We use Fisher classifier for classification and implement the algorithms for many times, and the averaged recognition rate is considered as the recognition accuracy. LPP, CLPP, and KCLPP [24] are implemented together with other feature extraction methods including PCA [5] and KDA [15] for comparison. As the results on ORL face database are shown in Table 6.16, the averaged

Table 6.16 Recognition accuracy on ORL subdatabases (%)

	PCA	KDA	LPP	CLPP	KCLPP	KSLPDA
ORL_SD1	92.00	93.50	95.00	96.00	96.50	98.50
ORL_SD2	92.00	93.50	93.50	94.00	95.50	97.50
ORL_SD3	93.00	94.00	95.50	97.00	98.50	99.50
ORL_SD4	92.00	93.50	93.50	94.50	96.00	97.50
ORL_SD5	90.50	91.00	91.50	92.50	96.00	97.00
Averaged	91.90	93.10	93.80	94.80	96.50	98.00

Table 6.17 Recognition accuracy on YALE subdatabases (%)

	PCA	KDA	LPP	CLPP	KCLPP	KSLPDA
YALE_SD1	84.33	94.44	86.33	90.00	94.44	95.67
YALE_SD2	86.77	92.22	90.67	91.11	92.22	93.33
YALE_SD3	85.33	93.33	88.56	86.67	93.33	94.44
YALE_SD4	85.67	93.33	88.89	90.00	93.33	92.33
YALE_SD5	88.67	96.67	95.56	93.33	96.67	97.44
Averaged	86.15	94.00	90.00	90.22	93.99	94.62

recognition rate of LPP, CLPP, and KCLPP and the proposed KSLPDA are 93.80 %, 94.80 %, 96.50 %, and 98.00 %, respectively. CLPP outperforms LPP owing to the class labels, KCLPP performs better than CLPP owing to kernel trick, and KSLPDA achieves the highest recognition accuracy through using kernel self-optimization and class information. Similar results obtained on the YALE database in Table 6.17 demonstrate that KSLPDA outperforms other algorithms on classification performance, and LPP, CLPP, KCLPP, and KSLPDA achieve the recognition accuracy of 90.00 %, 90.22 %, 93.99 %, and 94.62 %, respectively.

References

1. Hu YC (2007) Fuzzy integral-based perception for two-class pattern classification problems. Inf Sci 177(7):1673–1686
2. Chen C, Zhang J, Fleischer R (2010) Distance approximating dimension reduction of Riemannian manifolds. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 40(1):208–217
3. Zhang T, Huang K, Li X, Yang J, Tao D (2010) Discriminative orthogonal neighborhood-preserving projections for classification. IEEE Trans Syst Man Cybern B Cybern 40(1):253–263
4. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces versus Fisherfaces: recognition using class specific linear projection. Trans. Pattern Analysis Mach Intell 19(7):711–720
5. Batur AU, Hayes MH (2001) Linear subspace for illumination robust face recognition. In: Proceedings of IEEE international conference on computer vision and pattern recognition, pp 296–301
6. Hastie T, Stuetzle W (1989) Principal curves. J American Stat Assoc 84(406):502–516

7. Chang KY, Ghosh J (2001) A unified model for probabilistic principal surfaces. IEEE Trans Pattern Anal Mach Intell 23(1):22–41
8. Graepel T, Obermayer K (1999) A stochastic self-organizing map for proximity data. Neural Comput 11(1):139–155
9. Zhu Z, He H, Starzyk JA, Tseng C (2007) Self-organizing learning array and its application to economic and financial problems. Inf Sci 177(5):1180–1192
10. Yin H (2002) Data visualization and manifold mapping using the ViSOM. Neural Netw 15(8):1005–1016
11. Tenenbaum JB, Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290(5500):2319–2323
12. Roweis ST, Saul LK (2000) Nonlinear dimensionality deduction by locally linear embedding. Science 290(5500):2323–2326
13. He X, Niyogi P (2003) Locality preserving projections. In: Proceedings of conference in advances in neural information processing systems, pp 585–591
14. He X, Yan S, Hu Y, Niyogi P, Zhang HJ (2007) Face recognition using Laplacianfaces. IEEE Trans Pattern Anal Mach Intell 27(3):328–340
15. Zheng Z, Yang F, Tan W, Jia J, Yang J (2007) Gabor feature-based face recognition using supervised locality preserving projection. Signal Proces 87(10):2473–2483
16. Zhu L, Zhu S (2007) Face recognition based on orthogonal discriminant locality preserving projections. Neurocomputing 70(7–9):1543–1546
17. Cai D, He X, Han J, Zhang HJ (2006) Orthogonal Laplacianfaces for face recognition. IEEE Trans Image Proces 15(11):3608–3614
18. Yu X, Wang X (2008) Uncorrelated discriminant locality preserving projections. IEEE Signal Process Lett 15:361–364
19. Feng G, Hu D, Zhang D, Zhou Z (2006) An alternative formulation of kernel LPP with application to image recognition. Neurocomputing 69(13–15):1733–1738
20. van Gestel T, Baesens B, Martens D (2010) From linear to non-linear kernel based classifiers for bankruptcy prediction. Neurocomputing 73(16–18), pp 2955–2970
21. Zhua Q (2010) Reformative nonlinear feature extraction using kernel MSE. Neurocomputing 73(16–18):3334–3337
22. Cheng J, Liu Q, Lua H, Chen YW (2005) Supervised kernel locality preserving projections for face recognition. Neurocomputing 67:443–449
23. Zhao H, Sun S, Jing Z, Yang J (2006) Local structure based supervised feature extraction. Pattern Recogn 39(8):1546–1550
24. Li JB, Pan JS, Chu SC (2008) Kernel class-wise locality preserving projection. Inf Sci 178(7):1825–1835
25. Veerabhadrappa M, Rangarajan L (2010) Diagonal and secondary diagonal locality preserving projection for object recognition. Neurocomputing 73(16–18), pp 3328–3333
26. Wang X, Chung F-L, Wang S (2010) On minimum class locality preserving variance support vector machine. Pattern Recogn 43(8):2753–2762
27. Zhang L, Qiao L, Chen S (2010) Graph-optimized locality preserving projections. Pattern Recogn 43(6):1993–2002
28. Wang J, Zhang B, Wang S, Qi M, Kong J (2010) An adaptively weighted sub-pattern locality preserving projection for face recognition. J Netw Comput Appl 33(3):323–332
29. Lu G-F, Lin Z, Jin Z (2010) Face recognition using discriminant locality preserving projections based on maximum margin criterion. Pattern Recogn 43(10):3572–3579
30. Hu D, Feng G, Zhou Z (2007) Two-dimensional locality preserving projections (2DLPP) with its application to palmprint recognition. Pattern Recogn 40(1):339–342
31. Xu Y, Feng G, Zhao Y (2009) One improvement to two-dimensional locality preserving projection method for use with face recognition. Neurocomputing 73(1–3):245–249
32. Zhi R, Ruan Q (2008) Facial expression recognition based on two-dimensional discriminant locality preserving projections. Neurocomputing 71(7–9):1730–1734
33. Pan JS, Li JB, Lu ZM (2008) Adaptive quasiconformal kernel discriminant analysis. Neurocomputing 71(13–15):2754–2760

34. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Netw 12(6):783–789
35. Li JB, Pan JS, Chen SM (2007) Adaptive class-wise 2DKPCA for image matrix based feature extraction. In: Proceedings of the 12th conference on artificial intelligence and applications, Taiwan, pp 1055–1061
36. Xiong H, Swamy MN, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Netw 16(2):460–474
37. Mika S, Ratsch G, Weston J, Schölkopf B, Muller KR (1999) Fisher discriminant analysis with kernels. In: Proceedings of IEEE international workshop neural networks for signal processing 4:41–48

Chapter 7
Kernel Semi-Supervised Learning-Based Face Recognition

7.1 Introduction

Semi-supervised learning methods attempt to improve the performance of a supervised or an unsupervised learning in the presence of side information. This side information can be in the form of unlabeled samples in the supervised case or pair-wise constraints in the unsupervised case. Most existing semi-supervised learning approaches design a new objective function, which in turn leads to a new algorithm rather than improving the performance of an already available learner. Semi-supervised learning-based classifier design is feasible to solve the above problem. Kernel learning-based semi-supervised classifier is feasible to enhance the performance of CAD. The popular semi-supervised learning algorithms include disagreement algorithms [1–4], LDS [5], CM [6], and other side-information-based supervised learning algorithm [7–9], and these semi-supervised learning algorithms are widely used in computer-aided diagnosis [10], content-based image retrieval [11], and speech processing [12]. Semi-supervised kernel classifier has its potential application, but kernel function and its parameter will influence the performance. In the past research, Huang [13], Wang [14], Chen [15] had proposed some kernel learning method-based parameter adjusting of kernel function, and Xiong [16] introduced the data-dependent kernel and applied it to change the kernel structure through changing the relative parameter of data-dependent kernel. Micchelli and Pontil [17] proposed the combination of basic kernel method for kernel function choosing.

Techniques to perform dimensionality reduction for high-dimensional data can vary considerably from each other due to, e.g., different assumptions about the data distribution or the availability of the data labeling. We categorize them as follows [18]. Unsupervised DR—Principal component analysis (PCA) is the most well-known one that finds a linear mapping by maximizing the projected variances. For nonlinear DR techniques, isometric feature mapping (Isomap) [19] and locally linear embedding (LLE) [20] both exploit the manifold assumption to yield the embeddings. And, to resolve the out-of-sample problem in Isomap and LLE, locality preserving projections (LPP) [21] are proposed to uncover the data

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_7,

manifold by a linear relaxation. Supervised DR—Linear discriminant analysis (LDA) assumes that the data of each class have a Gaussian distribution, and derives a projection from simultaneously maximizing the between-class scatter and minimizing the within-class scatter. Alternatively, marginal Fisher analysis (MFA) [22] and local discriminant embedding (LDE) [23] adopt the assumption that the data of each class spread as a submanifold and seek a discriminant embedding over these submanifolds. Semi-supervised DR—If the observed data are partially labeled, dimensionality reduction can be performed by carrying out discriminant analysis over the labeled ones while preserving the intrinsic geometric structures of the remaining. Such techniques are useful, say, for vision applications where user interactions are involved, e.g., semi-supervised discriminant analysis (SDA) [24] for content-based image retrieval with relevance feedback.

The advantage of this method is that the input data structure can be optimized through self-adjusting the parameter of data-dependent kernel. In order to increase the generalization ability of semi-supervised kernel classifier, it is necessary to make the kernel to change with the different input training samples. So, it is necessary to study the kernel self-adaptive optimization for semi-supervised kernel learning classifier to improve the generalization ability of the classifier and to enhance the application system.

Three classical problems in pattern recognition and machine learning, namely, classification, clustering, and unsupervised feature selection, are extended to their semi-supervised counterpart. Unlabeled data are available in abundance, but it is difficult to learn the underlying structure of the data. Labeled data are scarce but are easier to learn from. Semi-supervised learning is designed to alleviate the problems of supervised and unsupervised learning problems and has gained significant interest in the machine learning research community.

Semi-supervised classification

Semi-supervised classification algorithms train a classifier given both labeled and unlabeled data. A special case of this is the well-known transductive learning [8], where the goal is to label only the unlabeled data available during training. Semi-supervised classification can also be viewed as an unsupervised learning problem with only a small amount of labeled training data.

Semi-supervised clustering

Clustering is an ill-posed problem, and it is difficult to come up with a general purpose objective function that works satisfactorily with an arbitrary dataset [4]. If any side information is available, it must be exploited to obtain a more useful or relevant clustering of the data. Most often, side information in the form of pairwise constraints ("a pair of objects belonging to the same cluster or different clusters") is available. The pairwise constraints are of two types: must-link and cannot-link constraints. The clustering algorithm must try to assign the same label to the pair of points participating in a must-link constraint and assign different labels to a pair

of points participating in a cannot-link constraint. These pairwise constraints may be specified by a user to encode his preferred clustering. Pairwise constraints can also be automatically inferred from the structure of the data, without a user having to specify them. As an example, web pages that are linked to one another may be considered as participating in a must-link constraint [9].

Semi-supervised feature selection

Feature selection can be performed for both supervised and unsupervised settings depending on the data available. Unsupervised feature selection is difficult for the same reasons that make clustering difficult—lack of a clear objective apart from the model assumptions. Supervised feature selection has the same limitations as classification, i.e., scarcity of labeled data. Semi-supervised feature selection aims to utilize pairwise constraints in order to identify a possibly superior subset of features for the task. Many other learning tasks, apart from classification and clustering, have their semi-supervised counterparts as well (e.g., semi-supervised ranking [10]). For example, page ranking algorithms used by search engines can utilize existing partial ranking information on the data to obtain a final ranking based on the query.

Although there are many algorithms on kernel learning-based semi-supervised classification methods, there is little research on improving the classifier through optimizing the kernel. This research is able to increase the theory research from other viewpoint for semi-supervised classification. This research expects to extend the research areas of kernel optimization. There are many theory researches on supervised learning-based kernel optimization, but there is little attention on the semi-supervised learning-based kernel optimization research. This research can extend the research area of kernel optimization and can be used in other kernel learning areas. This research supplies the theory support for other machine learning. The research ideas of kernel self-adaptive learning is introduced to other machine learning methods including nonlinear kernel discriminant analysis, support vector machine, and neural networks. Finally, this theoretical research fruits of semi-supervised learning can be used to solve the practical system problems including biometrics, medical image processing, and other classification systems.

Machine learning methods are divided into three kinds: supervised learning, unsupervised learning, and semi-supervised learning. Supervised learning—Given a set of input objects and a set of corresponding outputs (class labels) for the object, supervised learning aims to estimate a mapping such that the output for a test object (that was not seen during the training phase) may be predicted with high accuracy. The algorithm must learn a function f that predicts whether the user will be interested in a particular document that has not yet been labeled. Unsupervised Learning. Given a set of objects, a similarity measure between pairs of objects, the goal of unsupervised learning is to partition the set such that the objects within each group are more similar to each other than the objects between groups. For example, given a set of documents, the algorithm must group the documents into categories based on their contents alone without any external labels. Unsupervised learning is popularly known as clustering.

Supervised learning expects training data that are completely labeled. On the other extreme, unsupervised learning is applied on completely unlabeled data. Unsupervised learning is a more difficult problem than supervised learning due to the lack of a well-defined user-independent objective. For this reason, it is usually considered an ill-posed problem that is exploratory in nature; that is, the user is expected to validate the output of the unsupervised learning process. Devising a fully automatic unsupervised learning algorithm that is applicable in a variety of data settings is an extremely difficult problem and possibly infeasible. On the other hand, supervised learning is a relatively easier task compared to unsupervised learning. The ease comes with an added cost of creating a labeled training set. Labeling a large amount of data may be difficult in practice with the following causes. Firstly, data labeling is expensive: human experts are needed to perform labeling. For example, experts need to be paid to label, or tools such as Amazon's Mechanical turk must be used. Secondly, data labeling has uncertainty about the level of detail: the labels of objects change with the granularity at which the user looks at the object. As an example, a picture of a person can be labeled as "person", or at a greater detail face, eyes, torso, etc. Thirdly, data labeling is difficult: sometimes objects must be subdivided into coherent parts before they can be labeled. For example, speech signals and images have to be accurately segmented into syllables and objects, respectively, before labeling can be performed. Fourthly, data labeling can be ambiguous: objects might have nonunique labeling or the labeling themselves may be unreliable due to a disagreement among experts. Data labeling uses limited vocabulary: typical labeling setting involves selecting a label from a list of prespecified labels which may not completely or precisely describe an object. As an example, labeled image collections usually come with a prespecified vocabulary that can describe only the images that are already present in the training and testing data.

7.2 Semi-Supervised Graph-Based Global and Local Preserving Projection

Side-information-based semi-supervised dimensionality reduction (DR) is widely used in many areas without considering the unlabeled samples, and however, these samples carry some important information for DR. For that purpose of enough usage of side-information and unlabeled samples, we present a novel DR method called Semi-supervised Graph-based Global and Local Preserving Projection (SGGLPP) through integrating graph construction with the specific DR process into one unified framework; SGGLPP preserves not only the positive and negative constraints but also the local and global structure of the data in the low-dimensional space. In SGGLPP, the intrinsic and cost graphs are constructed using the positive and negative constraints from side information and k-nearest neighbor criterion from unlabeled samples. Experiments are implemented in two real image databases to test the feasibility and the performance of the proposed algorithm.

With the rapid accumulation of high-dimensional data, such as image classification, gene microarrays, dimensionality reduction (DR) method plays a more and more important role in practical data processing and analysis tasks. Graph-based DR methods have been successfully used as an important methodology in machine learning and pattern recognition fields. Recently, some semi-supervise graph-based DR methods are investigated, but they are based on side information, which only knows whether two samples come from the same classes or different classes but without knowing the class label information. All these algorithms use only the positive constraint information but ignore the unlabeled data.

In order to use the side-information and unlabeled samples, we propose a novel dimensionality reduction method namely Semi-supervised Graph-based Global and Local Preserving Projection (SGGLPP) for image classification. In SGGLPP, we construct the intrinsic and cost graphs using the positive and negative constraints with side-information samples, and we also apply the nearest neighbor graph with the unlabeled information samples.

We present the SGGLPP algorithm with graph-based viewpoint. Firstly, we introduce the side-information-based positive and negative constraint graph for the construction of the constraint equation to seek the projection matrix for dimensionality reduction. Secondly, in order to use enough the unlabeled samples, we introduce the side-information- and k-nearest neighbor-based positive and negative constraint graphs for solving the projection matrix. About the graph, the intrinsic graph demonstrates the relation between two samples belonging to the same classes, and the cost graph demonstrates the different classes of samples which are denoted by the constraint or the non-k-nearest neighbor.

7.2.1 Side-Information-Based Intrinsic and Cost Graph

We consider the side information of samples for training. The intrinsic graph is created by positive constraint, and the cost graph is constructed with the negative constraint. As shown in Fig. 7.1, we expect to shorten the distance of data points in intrinsic graph and lengthen the distance between the data points in the cost graph.

The mathematic expression of the intrinsic and cost graphs is shown as follows. Supposed that P is the positive constraint and N is the negative constraint,

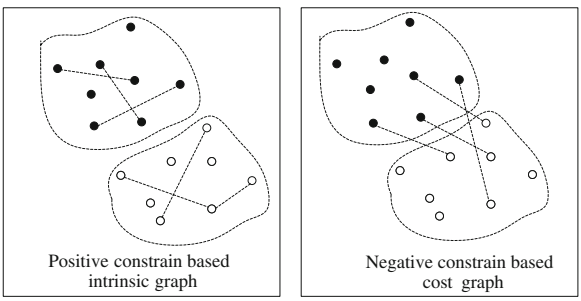


Fig. 7.1 Side-information-based intrinsic and cost graphs

P denotes that two samples belong to the same class, but the class labels are not known, and N denotes that two samples belong to two different classes of samples. From P and N, we cannot know the class labels of samples. Within-class compactness of samples M is defined as

$$\begin{aligned} M &= \sum_{(x_i,x_j)\in P} (w^T x_i - w^T x_j)^2 \\ &= 2\sum_i (w^T x_i D_{ii}^P x_i w) - 2\sum_{ij} (w^T x_i S_{ij}^P x_j w) \\ &= 2w^T X(D^P - S^P)X^T w \\ &= 2w^T X L^P X^T w \end{aligned} \tag{7.1}$$

where $S^P = \begin{cases} 1 & (x_i, x_j) \in P \\ 0 & \text{else} \end{cases}$, $D_{ii}^p = \sum_j S_{ij}^P$, and $L^P = D^P - S^P$.

Between–class compactness of samples is defined as

$$\begin{aligned} B &= \sum_{(x_i,x_j)\in N} (w^T x_i - w^T x_j)^2 \\ &= 2\sum_i (w^T x_i D_{ii}^N x_i w) - 2\sum_{ij} (w^T x_i S_{ij}^N x_j w) \\ &= 2w^T X(D^N - S^N)X^T w \\ &= 2w^T X L^N X^T w \end{aligned} \tag{7.2}$$

where $S^N = \begin{cases} 1 & (x_i, x_j) \in N \\ 0 & \text{else} \end{cases}$, $D_{ii}^N = \sum_j S_{ij}^N$, and $L^N = D^N - S^N$.

The objective function is defined as

$$w^* = \arg\max_w \frac{B}{M} = \arg\max_w \frac{w^T X L^P X^T w}{w^T X L^N X^T w} \tag{7.3}$$

The above equation aims to maximize the distance between the samples belonging to the different classes but minimize the distance within the samples from the same classes. This equation only considers the side information but without using the nonlabeled data.

7.2.2 Side-Information and k-Nearest Neighbor-Based Intrinsic and Cost Graph

In order to make enough use of a large number of unlabeled samples, we regard the samples belonging to the k-nearest neighbor as the positive constraint and the negative constraint for the non-k-nearest neighbor as shown in Fig. 7.2. Supposed that the samples will be closed in the low-dimensional projection space where its samples are closed in the high-dimensional input space. The improved objective function is defined as

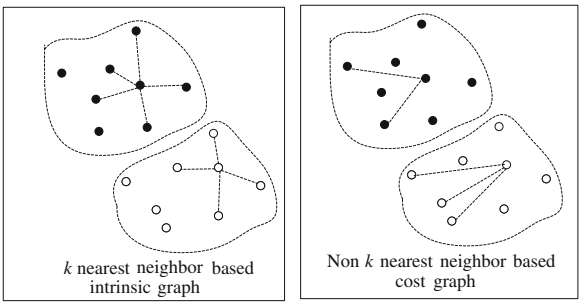


Fig. 7.2 k- and non-k-nearest neighbor-based intrinsic and cost graphs

$$w^* = \arg\max_w \frac{B + \alpha B_{\text{knn}}}{M + \beta M_{\text{knn}}} = \arg\max_w \frac{w^T X L^P X^T w + \alpha w^T X L^P_{\text{knn}} X^T w}{w^T X L^N X^T w + \beta w^T X L^N_{\text{knn}} X^T w} \tag{7.4}$$

where $B_{\text{knn}} = w^T X L^P_{\text{knn}} X^T w$, $M_{\text{knn}} = w^T X L^N_{\text{knn}} X^T w$. The L^P_{knn} and L^N_{knn} are calculated with the positive and negative graph created by the k-nearest neighbor criterion. The samples belong to the k-nearest neighbors of each other. α and β are weight parameters.

7.2.3 Algorithm Procedure

Input: The input samples set $X = x_1, x_2, \ldots, x_n \in R^{D \times n}$, positive constraint P and negative constraint N.
Output: Transformation matrix $W \in R^{D \times d} (d \prec D)$.

Step 1. Implement PCA to reduce the dimension of the training samples.
Step 2. Construct the positive and negative graph according to positive constraint P and negative constraint N and k- and non-k-nearest neighbor.
Step 3. Determine the weight parameters α and β, and obtain the transformation vector w^* through solving the constraint equation created with graph.
Step 4. Calculate the linear transformation matrix $W = w_1, w_2, \ldots, w_d$, where $w_1, w_2, \ldots, w_d$ are the eigenvectors corresponding to the d largest eigenvalues.
Step 5. Reduce the input vector x to the low-dimensional vector y with $y = W^T x$.

7.2.4 Simulation Results

In order to testify the feasibility and performance of SGGLPP algorithm, we construct the framework of experiment system. For the performance evaluation, we classify a new sample from the test samples and use the positive constraint to determine whether the classification result is right. We apply the image

classification as the practical application, and we use the face images to test the performance of the proposed algorithm. For the image classification task, face image classification is a difficult task owing to the pose, illumination, and expression changes. The practical databases, ORL and Yale databases, are used in the real system example.

In the experiments, we randomly choose five samples from the databases as the training samples. For semi-supervised learning, we construct the positive and negative constraints with 3 of these 5 samples according to the labeled class information. And the rest two samples are considered the unlabeled training samples. In the experiments, we construct positive and negative graphs with k- and non-k-nearest neighbor criterion. And the rest samples from the databases are considered as the test samples. We use the positive graph to evaluate the classification performance of the compared algorithm. For the purpose of comparison, we also implement the other semi-supervised and unsupervised learning algorithms, including principal component analysis (PCA), locality preserving projection (LPP), and semi-supervised locality preserving projection (SLPP). Since there is no labeled class information, we have not implemented the supervised learning for the comparison in the experiments for comparison. The experiment parameters including the weight parameter α, β, and k are determined with the cross-validation method; moreover, other algorithms achieve their optimal procedural parameters with cross-validation method.

As shown in Tables 7.1 and 7.2, SGGLPP outperforms other algorithms for the same training and test sets. Although the recognition rate is not high, it indicates that it is feasible to improve the performance with unlabeled classes together with the side information. Moreover, semi-supervised learning methods perform better than the unsupervised learning methods; for example, SLPP outperforms LPP.

SGGLPP integrates graph construction with specific SGGLPP-based DR process into a unified framework for image classification. The global and local structure of data is preserved in the low-dimensional feature space. In SGGLPP, we apply the graph-based constraint equation to seek the optimal projection matrix for dimensionality reduction. The intrinsic and cost graphs are constructed with the positive and negative constraints and k-nearest neighbor criterion. Experiments show that the SGGLPP algorithm outperforms other side-information-based semi-supervised learning such as SLPP and unsupervised learning such as PCA and LPP, which demonstrates that it is feasible to improve the performance with unlabeled samples using k-nearest neighbor graph. Besides the above advantage, the parameter selection is still one problem. The proposed algorithm chooses the

Table 7.1 Performance comparison on Yale face database

Algorithms	Recognition accuracy (%)
PCA	80
LPP	84
SLPP	86
SGGLPP	90

Table 7.2 Performance comparison on ORL face database

Algorithms	Recognition accuracy (%)
PCA	86
LPP	88
SLPP	89
SGGLPP	92

experimental parameters with the cross-validation method. Then, how to choose the procedural parameter is one important work in the future research.

7.3 Semi-Supervised Kernel Learning

The important ideas of this research on semi-supervised kernel optimized learning are shown as follows. (1) This research is to study small dataset-based object function of kernel optimization design methods. Compared with the supervised kernel learning, the loosing of the class labels of the samples is the problem endured by kernel object function design. This project is to propose the semi-supervised kernel learning, ensuring the generalization ability of the system. The object function is constructed to optimize the global ability by analyzing the relation between data distribution and class labels. (2) This research is to changeless structure-based semi-supervised kernel parameter. When the kernel function is selected in advance for some applications, in order to achieve the class discriminative ability, kernel self-adaptive optimization aims to solve the optimal kernel parameter according to the optimal object function designed by the loosing of the class labels information of samples. (3) This research is to changeable structure-based semi-supervised kernel self-adaptive optimization method. Owing to the complex data distribution and less priori knowledge in many applications, it cannot determine the kernel function in advance fitted to the sample data distribution. In order to make the kernel function to fit to the sample data distribution, this project extends the data-dependent kernel function under the semi-supervised learning and studies the optimization algorithm of seeking the parameter of data-dependent kernel. (4) This research is to construct the semi-supervised kernel self-adaptive optimization-based medical image recognition demo. The demo is constructed on the basis of practical computer-aided diagnosis system; the practical medical images are used to evaluate the system. And finally, the general application framework is presented for the semi-supervised kernel self-adaptive optimization.

The research framework is shown in Fig. 7.3. Firstly, given the training samples, we determine the label information (class labels information, side information). If the label information of training samples is given with the class label information, then the Fisher criterion and maximum margin criterion are applied to design the object function, and if the information of the training samples is given with the side information, then the global manifold preserving criterion is applied

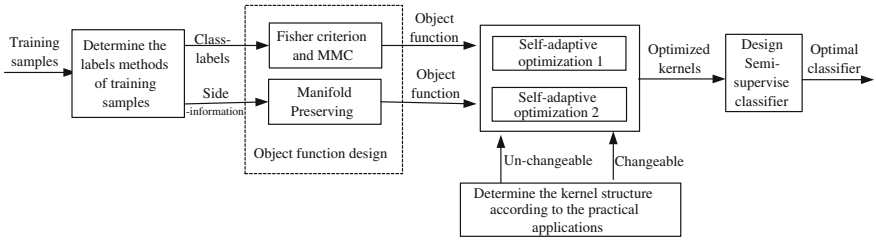


Fig. 7.3 Study frame of semi-supervised kernel self-adaptive optimization

to design the object optimization function. Secondly, given the object optimization function, according to the function of basic kernel, we apply the unchangeable kernel structure with the self-adaptive optimization method 1. And if we apply the changeable kernel structure, then we apply the data-dependent kernel and the self-adaptive optimization method 2 to solve the optimal parameter of data-dependent kernel to achieve the optimal kernel structure. Finally, we use the optimized kernel function to design the semi-supervised classifier to obtain the semi-supervised classifier.

Small labeled samples dataset-based objective function designing for kernel optimization

We study class label information and side-information-based kernel object function design. We apply the same structure mapping of the data, in the side information and class label information, and introduce empirical feature space. And we design the object function of kernel optimization in the empirical feature space. (1) Class label-based kernel optimization function design. This project is to use the Fisher criterion and maximum margin criterion as the design criterion for designing the kernel optimization function. Firstly, we use EM algorithm to estimate the class labels of unlabeled samples, and then we implement the optimization design. (2) Side-information-based kernel optimization function design. Firstly, for the input dataset, we construct the intrinsic graph and penalty graph and k-nearest neighbor graph, and then design the optimization function. For the input samples and the positive and negative constraints, where two samples come from the same class, they belong to positive constraint. Accordingly, they belong to negative constraint if they come from the different classes. The optimization equation is designed to preserve the side information of positive and negative constraints. In order to take enough use of unlabeled samples, the *k*-nearest neighbor and non-k-nearest neighbor graphs are constructed for the optimization equation.

Fisher criterion-based data-dependent kernel parameter optimization

The main idea is shown in Fig. 7.4. Firstly, we use the class label information-based data-dependent kernel, then we introduce the relations between data-dependent kernel and basic kernel, and create the function of parameter of

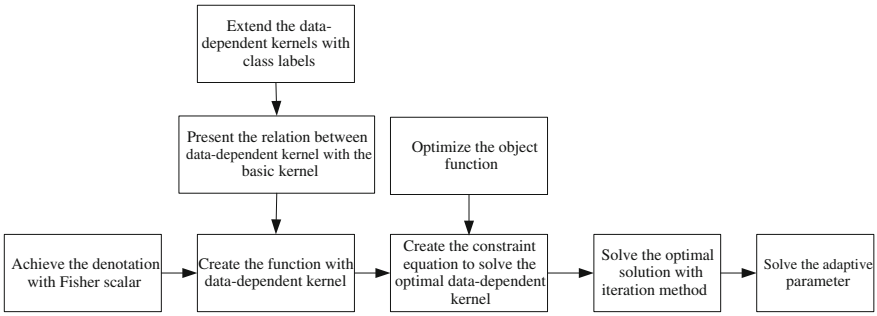


Fig. 7.4 Fisher criterion-based kernel optimization procedure

data-dependent kernel. Secondly, we design the object function according to the input samples and solve the constraints with data-dependent kernel. Finally, we use the iteration method to solve the optimal parameter according to the input sample.

Side-information-based changeable structure kernel self-adaptive optimization method

The main idea is shown in Fig. 7.5. Firstly, given the training samples, we determine the label information (class label information, side information). If the label information of training samples is given with the class label information, then the Fisher criterion and maximum margin criteria are applied to design the object function, and if the information of the training samples is given with the side information, then the global manifold preserving criterion is applied to design the object optimization function. Secondly, given the objection optimization function, according to the function of basic kernel, we apply the unchangeable kernel structure

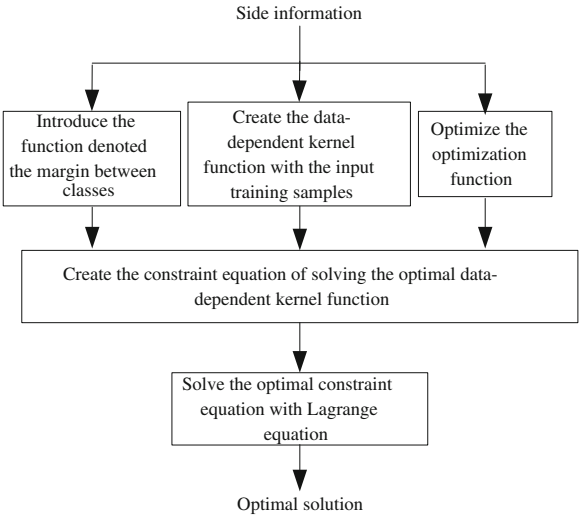


Fig. 7.5 Maximum margin criterion-based data-dependent kernel optimization procedure

with the self-adaptive optimization method 1. And if we apply the changeable kernel structure, we apply the data-dependent kernel and the self-adaptive optimization method 2 to solve the optimal parameter of data-dependent kernel to achieve the optimal kernel structure. Finally, we use the optimized kernel function to design the semi-supervised classifier to obtain the semi-supervised classifier.

Firstly, we use side information of training samples to introduce the function in the empirical feature space, and then we use side information to data-dependent kernel function to design the object function. Secondly, according the margin of samples, we create the data-dependent kernel and object function, and then create the constrain equation of data-dependent kernel. Finally, we use Lagrange method to solve the optimization equation.

Although the proposed research will mainly make theoretic and methodological contributions, we plan to investigate possible applications as well. One main domain of application is biometrics including 2D/3D face recognition, fingerprint recognition, iris recognition, and other biometrics. The research algorithms are applied into the practical 2D/3D recognition. The system framework is shown in Fig. 7.6.

7.3.1 Ksgglpp

The main idea of KSGGLPP is to map the original training samples to the feature space F through the nonlinear mapping Φ and then to implement linear discriminant analysis in the feature space F. Supposed N-dimensional M training sample $\{x_1, x_2, \ldots, x_M\}$ from L classes, then

$$\Phi : \mathbb{R}^N \rightarrow F, x \mapsto \Phi(x) \tag{7.4}$$

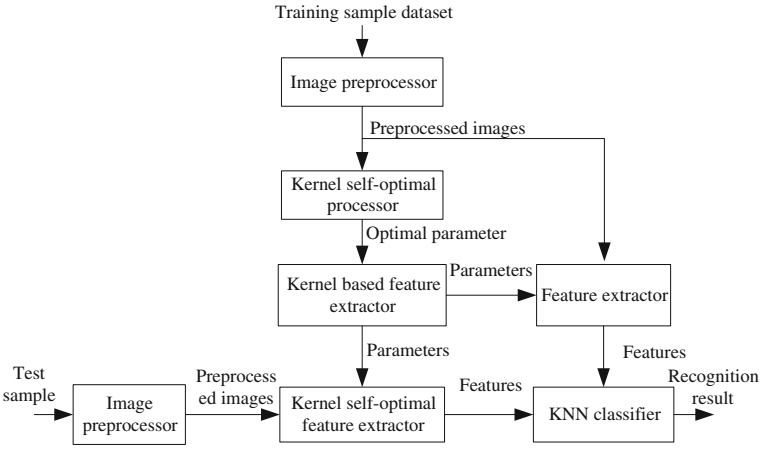


Fig. 7.6 Framework of semi-supervised kernel self-optimization-based medical face recognition

The dimension of feature space F is very high, in order to avoid to deal with the mapped samples, kernel function is calculated by $k(x,y) = <\Phi(x), \Phi(y)>$. KSGGLPP is shown in $\Phi(X) = [\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_n)]$ in the Hilbert space. The objective function of KSGGLPP in the Hilbert space is written as follows.

$$w^{\Phi *} = \arg\max_{w^{\Phi}} \frac{B^{\Phi}}{M^{\Phi}} = \arg\max_{w^{\Phi}} \frac{w^{\Phi T} X^{\Phi} L^{\Phi P} X^{\Phi T} w^{\Phi}}{w^{\Phi T} X^{\Phi} L^{\Phi N} X^{\Phi T} w^{\Phi}} \tag{7.5}$$

where S_{ij}^{Φ} is a similarity matrix with weights characterizing the likelihood of two points in the Hilbert space, and z_i^{Φ} is the one-dimensional representation of $\Phi(x_i)$ with the a projection vector w^{Φ}, i.e.,$z_i^{\Phi} = \left(w^{\Phi}\right)^T \Phi(x_i)$. The optimal transformation matrix w^{Φ} is calculated through solving an eigenvalue problem. Any eigenvector may be expressed by a linear combination of the observations in feature space as follows.

$$w^{\Phi} = \sum_{p=1}^{n} \alpha_p \Phi\left(x_p\right) = Q\alpha \tag{7.6}$$

where $Q = [\Phi(x_1) \quad \Phi(x_2) \quad \cdots \quad \Phi(x_n)]$ and $\alpha = [\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_n]$. We formulate SGGLPP with the dot product to generalize it to the nonlinear case. The dot product in the Hilbert space is presented with a kernel function, i.e., $k\left(x_i, x_j\right) = \Phi(x_i)^T \Phi\left(x_j\right)$. Matrix D^{Φ} provides a natural measure on the data points. The bigger the value D_{ii}^{Φ} (corresponding to z_i^{Φ}) is, the more "important" is z_i^{Φ}. Then minimization problem in (4) is transformed as

$$\alpha^* = \arg_{\alpha}\max \frac{\alpha^T K L^{\Phi P} K \alpha}{\alpha^T K L^{\Phi N} K \alpha} \tag{7.7}$$

$D_{ii}^{\Phi N} = \sum_j S_{ij}^{\Phi N}$, and $L^{\Phi N} = D^{\Phi N} - S^{\Phi N}$. Now let us consider QR decomposition of matrix K, i.e., $K = P\Lambda P^T$, where $P = [r_1, r_2, \ldots, r_m]$, $(m \leq n)$, and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$, and $r_1, r_2, \ldots, r_m$ are K's orthonormal eigenvector corresponding to m largest nonzero eigenvalue $\lambda_1, \lambda_2, \ldots, \lambda_m$. Let $\beta = \Lambda^{\frac{1}{2}} P^T \alpha$, i.e., $\alpha = \Lambda^{-\frac{1}{2}} P^T \beta$, we can reconsider Eq. (7.6) as follows.

$$w^{\Phi} = Q\alpha = QP\Lambda^{-\frac{1}{2}}\beta \tag{7.8}$$

Then z_i^{Φ} is the one-dimensional representation of $\Phi(x_i)$ with the a projection vector w^{Φ}, i.e.,.

$$z_i^{\Phi} = \left(QP\Lambda^{-\frac{1}{2}}\right)^T \Phi(x_i) \tag{7.9}$$

Consequently, $z_i^{\Phi} = \beta^T y_i$, where

$$\begin{aligned} y_i &= \left(QP\Lambda^{-\frac{1}{2}}\right)^T \Phi(x_i) \\ &= \left(\frac{r_1}{\sqrt{\lambda_1}}\frac{r_2}{\sqrt{\lambda_2}}\cdots\frac{r_m}{\sqrt{\lambda_m}}\right)^T [k(x_1,x),k(x_2,x),\ldots,k(x_n,x)]^T \end{aligned} \tag{7.10}$$

where $r_1, r_2, \ldots, r_m$ are K's orthonormal eigenvector corresponding to m largest nonzero eigenvalue $\lambda_1, \lambda_2, \ldots, \lambda_m$. The procedure of KSGGLPP is divided into three steps. Project x_i to y_i with the KPCA transformation by keeping the 98 percent information. Construct the similarity matrix S^Φ with kernel function. (3) Project y_i to the KSGGLPP transformed feature z_i^Φ with $z_i^\Phi = \left(W_{\text{KSGGLPP}}^\Phi\right)_{y_i}^T$, where W_{KSGGLPP}^Φ is a matrix whose column vectors are the d eigenvectors corresponding to the first d smallest eigenvalues.

7.3.2 Experimental Results

On six UCI datasets, we implement some algorithms to compare the unsupervised learning (PCA, LPP), supervised learning (SLPP [18]), and semi-supervised learning (SGGLPP, KSGGLPP). We use the whole unlabeled training samples and labeled training samples. In the experiments, we choose the Gaussian kernel with its parameters determined by the training samples. For the purpose of comparison, we also implement the other semi-supervised and unsupervised learning algorithms, including principal component analysis (PCA), locality preserving projection (LPP), and supervised locality preserving projection (SLPP) [18]. The comparison of the results shows KSGGLPP achieves the highest recognition accuracy than SGGLPP, SLPP, LPP, and PCA. In these algorithms, PCA and LPP apply the whole training samples with unsupervised learning, and KSGGLPP and SGGLPP use the whole training samples with semi-supervised learning.

As shown in Tables 7.2 and 7.3, SGGLPP outperforms other algorithm for the same training and test sets. Although the recognition rate is not high, it indicates that it is feasible to improve the performance with unlabeled classes together with the side-information. Moreover, semi-supervised learning methods perform better than the unsupervised learning methods, for example, SLPP outperforms LPP.

Table 7.3 Recognition performance of KPCA error rate (%)

Datasets	Training samples/ labeled samples	PCA	LPP [18]	SLPP [18]	SGGLPP	KSGGLPP
Banana	400/120	14.5 ± 0.1	14.2 ± 0.2	14.0 ± 0.2	13.7 ± 0.2	13.4 ± 0.1
Image	1,300/180	5.6 ± 0.2	5.3 ± 0.2	4.7 ± 0.3	4.5 ± 0.3	4.3 ± 0.4
F.Solar	666/50	35.2 ± 2.1	34.2 ± 2.3	32.1 ± 2.4	29.9 ± 2.2	29.4 ± 2.3
Splice	1,000/280	9.0 ± 0.8	8.8 ± 0.7	8.7 ± 0.5	8.5 ± 0.6	8.2 ± 0.7
Thyroid	140/30	2.5 ± 1.1	2.4 ± 1.0	2.3 ± 1.0	2.2 ± 1.2	2.0 ± 1.1
Titanic	150/30	24.8 ± 0.8	24.4 ± 0.9	23.3 ± 0.4	22.3 ± 0.3	21.8 ± 0.2

Table 7.4 Performance comparison on Yale face database

Algorithms	Recognition accuracy (%)
PCA	80
LPP [18]	84
SLPP [18]	86
SGGLPP	90
KSGGLPP	94

Table 7.5 Performance comparison on ORL face database

Algorithms	Recognition accuracy (%)
PCA	86
LPP [18]	88
SLPP [18]	89
SGGLPP	92
KSGGLPP	96

Moreover, KSGGLPP performs best among these algorithms, which denotes the kernel method is feasible to improve the performance of SGGLPP. In the experiments, kernel functions and its parameters are chosen with cross-validation methods. Gaussian kernel and polynomial kernel are chosen as the candidate kernel for kernel learning (Tables 7.4, 7.5).

References

1. Zhou ZH, Li M (2005) Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 17(11):1529–1541
2. Zhou Z (2007) The co-training paradigm in semi-supervised learning. Machine learning and applications. Tsinghua University Press, Beijing, 259–275 (in Chinese)
3. Zhou Z (2008) Semi-supervised learning by disagreement. Proceedings of the 4th conference on granular computing. Wiley-IEEE Press, Hoboken
4. Blum A, Mitchell T (1998) Combining labeled and unlabeled data wit h co-training. Proceedings of the 11th annual conference on computational learning theory. ACM, New York, pp 92–100
5. Chapelle O, Zien A (2005) Semi-supervised classification by low density separation. Proceedings of the 10th international workshop on artificial intelligence and statistics. Pascal Press, Savannah, Barbados, pp 245–250
6. Zhou D, Bousquet O, Thomas N (2003) Learning with local and global consistency. Max Planck Institute for Biological Cybernetics, pp 154–157
7. Xing EP, Jordan MI, Russell S (2003) Distance metric learning with application to clustering with side-information. Advance in neural information processing systems, MIT Press, Cambridge, pp 505–512
8. Tang W, Zhong S (2006) Pairwise constraints-guided dimensionality reduction. Proceedings of data mining workshop on feature selection for data mining. pp 59–66

9. Yeung DY, Chang H (2006) Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. Pattern Recogn 39(5):1007–1010
10. Li M, Zhou ZH (2007) Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. IEEE Trans Syst, Man and Cybernetics-Part A, 37(6):1088–1098
11. Zhou ZH, Chen KJ, Dai HB (2006) Enhancing relevance feedback in image retrieval using unlabeled data. ACM Trans Inf Syst 24(2):219–244
12. Steedman M, Osborne M, Sarkar A (2003) Bootstrapping statistical parsers from small datasets. Proceedings of the 11th conference on the european chapter of the association for computational linguistics, 331–338
13. Huang J, Yuen P, Chen WS, Lai JH (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. Proceedings of the sixth IEEE international conference on automatic face and gesture recognition, pp 1352–1355
14. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst, Man and Cybern-Part B: Cybern 35(3):556–562
15. Chen WS, Yuen P, Huang J, Dai DQ (2005) Kernel machine-based one-parameter regularized fisher discriminant method for face recognition. IEEE Trans Syst, Man and Cybern-Part B: Cybern 35(4):658–669
16. Xiong HL, Swamy MN, Ahmad M (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Networks 16(2):460–474
17. Charles A, Massimiliano P (2005) Learning the kernel function via regularization. J Machine Learn Res 6:1099–1125
18. Lin Y–Y, Liu T-L, Fuh C-S (2011) Multiple kernel learning for dimensionality reduction. IEEE Trans Pattern Anal Machine Intell 33(6):1147–1160
19. Tenenbaum J, de Silva V, Langford J (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290:2319–2323
20. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323–2326
21. He X, Niyogi P (2003) Locality preserving projections, advances in neural information processing systems. MIT Press, Cambridge
22. Yan S, Xu D, Zhang B, Zhang H, Yang Q, Lin S (2007) Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Trans Pattern Anal Machine Intell 29(1):40–51
23. Chen H-T, Chang H-W, Liu T-L (2005) Local discriminant embedding and its variants. Proc IEEE Conf Comput Vision and Pattern Recognition, pp 846–853
24. Cai D, He X, Han J (2007) Semi-supervised discriminant analysis, Proc IEEE Int'l Conf Computer Vision

Chapter 8
Kernel-Learning-Based Face Recognition for Smart Environment

8.1 Introduction

Multimedia multisensor system is used in various monitoring systems such as bus, home, shopping mall, school, and so on. Accordingly these systems are implemented in an ambient space. Multiple sensors such as audio and video are used for identification and ensure the safety. The wrist pulse signal detector is used to health analysis. These multisensor multimedia systems are be recording, processing, and analyzing the sensory media streams and providing the high-level information.

Person identification using face, palmprint, and fingerprint images is crucial in smart environment. Extracting the features of the images is the important step for classification. As the dimensionality reduction methods for feature extraction, principal component analysis (PCA) and linear discriminant analysis (LDA) methods are widely used successfully in real-world applications. In order to overcome this weakness of LDA, the kernel trick is used to represent the complicated nonlinear relationships of input data to develop kernel discriminant analysis (KDA) algorithm. Kernel-based nonlinear feature extraction techniques have attracted much attention in the areas of pattern recognition and machine learning [1, 2]. Some algorithms using the kernel trick are developed in recent years, such as kernel principal component analysis (KPCA) [3], kernel discriminant analysis (KDA) [4], and support vector machine (SVM) [5]. KDA has been applied in many real-world applications owing to its excellent performance on feature extraction. Researchers have developed a series of KDA algorithms [6–26]. Because the geometrical structure of the data in the kernel mapping space, which is totally determined by the kernel function, has significant impact on the performance of these KDA methods. The separability of the data in the feature space could be even worse if an inappropriate kernel is used. However, choosing the parameters for kernel just from a set of discrete values of the parameters does not change the geometrical structures of the data in the kernel mapping space. In order to overcome the limitation of the conventional KDA, we introduce a novel kernel named quasiconformal kernel which were widely studied, where the geometrical

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_8,

structure of data in the feature space is changeable with the different parameters of the quasiconformal kernel.

All these methods process the image as the vector through transforming the image matrix into vector. This procedure causes the high dimensionality problem, and accordingly, it also is difficult to implement PCA, LDA, and other similar dimensionality reduction. For example, one image of 100 plus 100 pixels as usually is low resolution image in many applications. The corresponding dimensionality of vector is 10,000. That is, the correlation matrix is 10,000 plus 10,000. It brings the high computation problem. So researchers developed the 2D PCA which reduces the computational cost by directly using the original image matrices. Current method endures the following problems, (1) 2D PCA is not suitable to nonlinear feature extraction, while currently there is no 2D kernel function to process the image matrix directly; (2) kernel learning is heavy influenced by parameter selection, the current parameter adjusting using the fixed kernel function cannot change the data structure; (3) the large size of training samples occurs the high computation and saving space for kernel matrix. In this work, we present a framework of multisensor multimedia data with kernel optimization-based 2D principal analysis for smart environment for identification and health analysis. The face, palmprint, and fingerprint images are used to identify the person, and the wrist pulse signal is used to evaluate the performance on health analysis. This framework uses 2D kernel function of processing the image matrix without vector transforming, applies proposed kernel optimization-based 2D kernel with adaptive self-adjusting parameter ability, and presents kernel optimization-based 2D principal analysis for image feature extraction for the higher computation efficiency and recognition performance.

Sensor-based monitoring systems use multiple sensors to identify high-level information based on the events that take place in the monitored environment. Identification and health care are the important tasks in the smart environment. This book presents a framework of multisensory multimedia data analysis using kernel optimization-based principal analysis for identification and health care in the smart environment. The face, palmprint, and fingerprint images are used to identify the persons, and the wrist pulse signal is to analyze the person's health conditions. The recognition performances evaluations are implemented on the complex dataset of face, palmprint, fingerprint, and wrist pulse signal. The experimental results show that the proposed algorithms perform well on identification and health analysis.

8.2 Framework

The framework of the multimedia multisensor smart environment is shown in Fig. 8.1. The sensors S1, S2, and Sn denote the different sensors to collect the different multimedia data. Each data is preprocessed or processed with media processor, and then, the features of the sensor data are extracted for event detection.

The event detectors include the identification and health analysis. So the multimedia multisensor decision is to determine the identification and health status of the entrance person. *Identification*: In this step, the face, fingerprint, and iris images are to identify the person with the fusion method. The face, fingerprint, and images are extracted with kernel optimization-based 2D principal analysis. *Health*: Wrist pulse signals with the vital information of health activities can reflect the pathologic changes of a person's body condition. The health conditions of a patient are detected with his wrist pulses in the smart environment. This method is widely used in traditional Chinese medicine for over thousands of years. Today modern clinical diagnosis also applies the arterial elasticity and endothelial function for patients with certain diseases, such as hypertension, hypercholesterolemia, and diabetes [27]. The wrist pulse signals are used to analyze a person's health status in that they reflect the pathologic changes of the body condition. Different practitioners may not give identical diagnosis results for the same patient. Developing computerized pulse signal analysis techniques with the digitized pulse signals [28–32] is to standardize and objectify the pulse diagnosis method. Traditional Chinese pulse diagnosis is widely paid attentions through the wrist pulse assessment. In this framework, the wrist pulse signals are analyzed with Gaussian model.

The architecture of the practical system is shown in Fig. 8.2. The practical application system contains user interface, computation module, fusion module, and data collection and processing module. In the user interface module, users apply the interface including identification and health for the persons entering the smart room. In the computation module, the kernel learning algorithm is applied into feature extraction of multimedia multisensor data. Fusion module implements the identification results and wrist pulse detection as the final results. The cameras and wrist pulse sensors are used to data collecting; and the health analysis service; face, iris, and fingerprint identification. The samples of signal, face, iris, and

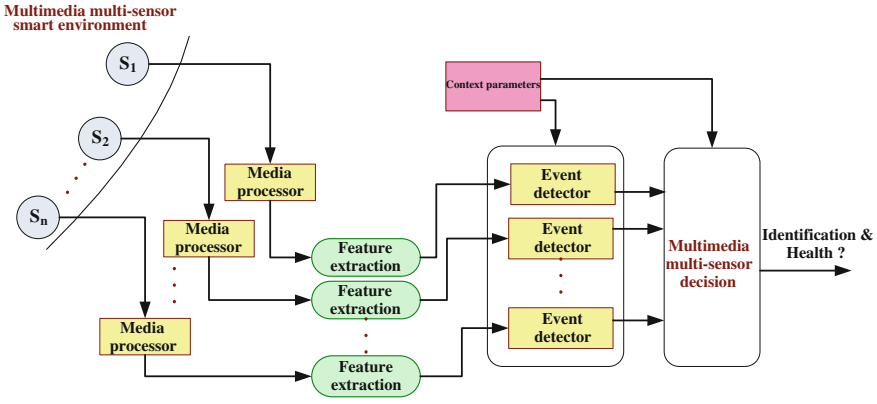


Fig. 8.1 Framework of generic event detection (identification and health) based multisensor multimedia streams

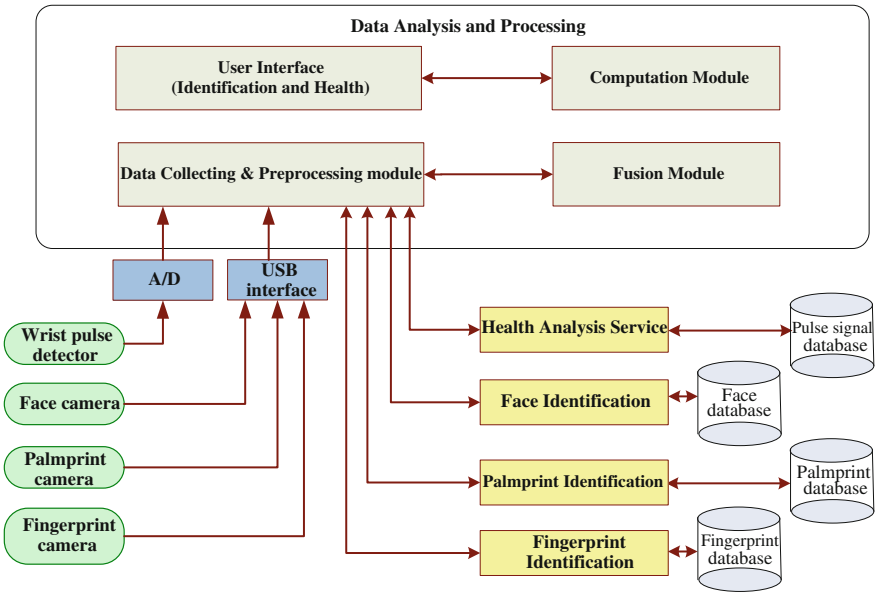


Fig. 8.2 Architecture of the application system

fingerprint images are saved in the four separate databases, and these samples are used to identification and health analysis.

Wrist pulse recognition: The wrist pulse signal of a person contains important information about the pathologic changes of the person's body condition. Extracting this information from the wrist pulse waveforms is important for computerized pulse diagnosis [33]. This book aims to establish a systematic approach to the computerized pulse signal diagnosis, with the focus placed on feature extraction and pattern classification. The collected wrist pulse signals are first denoised by the wavelet transform. To effectively and reliably extract the features of the pulse signals, a two-term Gaussian model is then adopted to fit the pulse signals. The reason of using this model is because each period of a typical pulse signal is composed of a systolic wave and a diastolic wave, both of which are bell-shaped [34].

Face recognition: Compared with the fingerprint, retina, iris, and other human biometric recognition system, face recognition system is more directly, user friendly, without any mental disorder, and can get some information though facial expressions, posture analysis which other recognition system difficult to get.

Palmprint recognition: Palmprint recognition is a new branch of biometric identification technology. Compared with other biometric technology, palmprint recognition has some unique advantages: palmprint information is not related to privacy issues; rich information but also has the uniqueness and stability of low cost acquisition equipment. Face and palmprint recognition both have not infringe on the user and acceptable levels are high. The most important is these two kinds

of biometric recognition method can use the same acquisition device. For example, using a low resolution of the camera can complete acquisition of face and palmprint image. At the same time, we can use a same method for feature extraction of face and palmprint recognition. So the features are compatible, we can analyze the integration at all levels. This book will draw lessons from the newest achievement of the international biometric recognition, information fusion and image processing, information fusion problem analysis. Research on the face and palmprint biological feature fusion based on the face recognition and palmprint recognition.

8.3 Computation

In the computation module, the two issues include feature extraction and classification. Firstly, the feature extraction uses dimensionality reduction method to extract the features of input multimedia multisensory data. Especially, on the wrist pulse signal, the wavelet decomposition method is to extract the features of the input pulse signal for health analysis. The features of the face, palmprint, and fingerprint images are extracted with 2D Gaussian-kernel-based dimensionality reduction for the identification. Secondly, the classification method is to determine the health/disease and identification through classifying the features.

8.3.1 Feature Extraction Module

For the large size of training sample $I = \{I_1, I_2, \ldots, I_M\}$, the N_z representative training samples $I' = \left\{I'_1, I'_2, \ldots, I'_{N_z}\right\}$ with sparse analysis, and then the optimized 2D kernel mapping $K^{(\alpha)}(X, Y)$ for input image matrix X and Y with the optimized parameter β, the feature Y of the input image matrix I_{input} is

$$y = WK^{(\alpha)}\left(I_{\text{input}}, I'_j\right), j = 1, 2, \ldots, N_z, \tag{8.1}$$

where the projection matrix W is solved under the different criterion. The method (8.1) is divided into two steps, one is to find the representative training samples $I' = \left\{I'_1, I'_2, \ldots, I'_{N_z}\right\}$ with sparse analysis, and second is to seek the optimized 2D kernel mapping $K^{(\alpha)}(X, Y)$ with the optimized parameter α.

Step 1. Finding the representative training samples

We use direct sparse kernel learning method using definitions of "expansion coefficients" and "expansion vectors." Suppose a matrix $I' = \left\{I'_1, I'_2, \ldots, I'_{N_z}\right\}$, composed of N_z expansion vectors, and β_i $(j = 1, 2, \ldots, N)$ $(N < M)$ are expansion coefficients, we modify the optimization problem to the following problem:

$$\begin{aligned} &\max_{w,e} J(w,e) = -\frac{1}{2}w^T w + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ &\text{subject to } e_i = w^T\left(\phi(I_i) - u^{\phi}\right),\ i = 1,2,\ldots,N, \\ &\qquad w = \sum_{i=1}^{N} \phi(I_i)\beta_i \end{aligned} \tag{8.2}$$

where $\phi(I) = \left[\phi(I_1), \phi(I_2), \ldots, \phi(I_{N_z})\right]$. Now our goal is to solve the above optimization problem, similar to the method in [35]. We can divide the above optimization problem into two steps, one is to find the optimal expansion vectors and expansion coefficients; second is to find the optimal projection matrix. Firstly, we reduce the above optimization problem, and then, we can obtain

$$\begin{aligned} &\max_{\beta,e} J(\beta,e) = -\frac{1}{2}\sum_{r=1}^{N}\sum_{s=1}^{N}\beta_s\beta_r\phi(I_r)^T\phi(I_s) + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ &\text{subject to } e_i = \left(\sum_{r=1}^{N}\beta_r\phi(I_r)^T\right)\left(\phi(I_i) - u^{\phi}\right) \end{aligned} \tag{8.3}$$

where $u^{\phi} = \left(\frac{1}{N}\right)\sum_{i=1}^{N}\phi(x_i)$. We apply the kernel function, that is, given a random I', then

$$\begin{aligned} &W = \max_{\beta,e} -\frac{1}{2}\beta^T K\beta + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \\ &\text{subject to } e_i = \beta^T g(I_i) \end{aligned} \tag{8.4}$$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_N]^T$, and $K_{ij} = k\left(I_i, I_j\right)$, and $g(I_i)$ is calculated with

$$g(I_i) = \left[k(I_n, I_i) - \frac{1}{N}\sum_{q=1}^{N} k(I_n, I_q)\right]^T_{n=1,2,\ldots,N} \tag{8.5}$$

The solution of the above-constrained optimization problem can often be found by using the so-called Lagrangian method.

Step 2. Seeking the optimized 2D kernel mapping

Based on the above definition, we extend it to the matrix version and propose the adaptive 2D Gaussian kernel as follows. Suppose that $k_b(X,Y)$ is so-called matrix-norm-based Gaussian kernel (M-Gaussian kernel) as the basic kernel and $k_d(X,Y)$ is a data-dependent matrix-norm-based Gaussian kernel.

It is easy to prove that $k(X,Y) = e^{-\frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}(x_{ij}-y_{ij})^2\right)^{1/2}}{2\sigma^2}}$ is a kernel function. Kernel function can be defined in various ways. In most cases, however, kernel means a function whose value only depends on a distance between the input data,

which may be vectors. And the following is the concept of Gram matrix and a sufficient and necessary condition for a symmetric function to be a kernel function.

Then data-dependent matrix-norm-based Gaussian kernel is defined as follows

$$k_d(X,Y) = f(X)f(Y)k_b(X,Y) \tag{8.6}$$

where $f(X)$ is a positive real-valued function X,

$$f(X) = b_0 + \sum_{n=1}^{N_{XV}} b_n e\left(X, \tilde{X}_n\right) \tag{8.7}$$

where $e\left(X, \tilde{X}_n\right)(1 \le n \le N_{\mathrm{XM}})$ is defined as follows:

$$e\left(X, \tilde{X}_n\right) = \exp\left(-\delta \sum_{j=1}^{N} \left(\sum_{i=1}^{M} \left(x_{ij} - \tilde{x_{ij}}\right)^2\right)^{1/2}\right) \tag{8.8}$$

where $\tilde{x_{ij}}(i = 1, 2, \ldots, M, \quad j = 1, 2, \ldots, N)$ is the elements of matrix $\tilde{X_n}$ $(n = 1, 2, \ldots, N_{\mathrm{XM}})$, and δ is a free parameter, and $\tilde{X}_n$, $1 \le n \le N_{\mathrm{XM}}$ are called the "expansion matrices (XMs)" in this book, N_{XM} is the number of XMs, and $b_i \in R$ is the "expansion coefficient" associated with $\tilde{X}_n$. The $\tilde{X}_n$, $1 \le n \le N_{\mathrm{XM}}$, for its vector version, have different notations in the different kernel learning algorithms. Given n samples X_p^q $(X_p^q \in \mathbb{R}^{M \times N})$, $(p = 1, 2, \ldots, L, \ q = 1, 2, \ldots, n_p)$ where n_p denotes the number of samples in the pth class and L denote the number of the classes. M-Gaussian kernel $k_b(X,Y)$ is defined as follows:

$$k(X,Y) = \exp\left(-\frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}\left(x_{\mathrm{ij}} - y_{\mathrm{ij}}\right)^2\right)^{1/2}}{2\sigma^2}\right)(\sigma \ > 0) \tag{8.9}$$

where $X = \left[x_{\mathrm{ij}}\right]_{i=1,2,\ldots,M;j=1,2,\ldots,N}$ and $Y = \left[y_{ij}\right]_{i=1,2,\ldots,M;j=1,2,\ldots,N}$ denote two sample matrices.

Now our goal is to optimize the kernel function to maximize the largest class discrimination in the feature space. We apply the maximum margin criterion to solve the expansion coefficients. We maximize the class discrimination in the high-dimensional feature space by maximizing the average margin between different classes which is widely used as maximum margin criterion for feature extraction. As the expansion vectors and the free parameter, our goal is to find the expansion coefficients varied with the input data to optimize the kernel. Given one free parameter δ and the expansion vectors $\{I_i\}_{i=1,2,\ldots,N_z}$, we create a matrix as

$$E = \begin{bmatrix} 1 & e\left(X_1, \tilde{X}_1\right) & \cdots & e\left(X_1, \tilde{X}_{N_{\mathrm{XMs}}}\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e\left(X_M, \tilde{X}_1\right) & \cdots & e\left(X_M, \tilde{X}_{N_{\mathrm{XMs}}}\right) \end{bmatrix} \tag{8.10}$$

Let $\beta = [b_0, b_1, b_2, \ldots, b_{N_z}]^T$ and $\Lambda = \mathrm{diag}(f(X_1), f(X_2), \ldots, f(X_{N_z}))$, we obtain

$$\Lambda 1_n = E\beta \tag{8.11}$$

So the problem of solving the constrained optimization function is transformed to the problem of solving eigenvalue equation. We can obtain the optimal expansion coefficient vector β^*, that is, the eigenvector of $E^T ME$ corresponding to the largest eigenvalue. It is easy to see that the data-dependent kernel with β^* is adaptive to the input data, which leads to the best class discrimination in feature space for given input data.

Given the input samples, we obtain the adaptive 2D kernel, where the samples have the largest class separability in the high kernel space. Then, we implement KPCA with adaptive 2D kernel for image feature extraction, which is so-called adaptive class-wise 2DKPCA for performance evaluation on the proposed scheme.

8.3.2 Classification

After feature extraction, the nearest neighbor (to the mean) classifier with the similarity measure δ is applied to classification:

$$\delta\left(F, M_k^0\right) = \min_j \quad \delta\left(F, M_j^0\right) \tag{8.12}$$

where M_k^0, $k = 1, 2, \ldots, L$, is the mean of the training samples for class ω_k. The feature vector F is classified as belonging to the class of the closet mean, M_k^0, using the similarity measure δ. Another classifier, such as Fisher classifier, is also used to classification, and we evaluate their performance in the experiments.

8.4 Simulation and Analysis

8.4.1 Experimental Setting

We have the simulations to testify the proposed framework in the complex database consisted of 50 persons. Each person has 10 samples, and each sample contains face image, palmprint image, fingerprint image, and wrist pulse signal. The face images come from ORL and Yale databases [36, 37], palmprint image [33], fingerprint image, wrist pulse signal from Hong Kong Polytechnic University [34]. Some examples are shown in Fig. 8.3. ORL face database, developed at the Olivetti Research Laboratory, Cambridge, U.K., is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time, and facial expression. The YALE face database was constructed at the YALE Center for Computational Vision and Control. It contains 165 grayscale images of

15 individuals. These images are taken under different lighting condition, and different facial expression, and with/without glasses. To reduce computation complexity, we resize the original ORL face images sized 112×92 pixels with a 256 gray scale to 48×48 pixels. Similarly, the images from YALE databases are cropped to the size of 100×100 pixels. The training set and test set are determined as shown in Table 8.1.

After describing the two dataset used in our experiments, it is worthwhile to make some remarks on the experiment setting as follows: (1) we run experiments for 10 times, and the average rate is used to evaluate the classification performance. (2) the experiments are implemented on a Pentium 3.0 GHz computer with 512 MB RAM and programmed in the MATLAB platform (Version 6.5). (3) the procedural parameters, i.e., kernel parameters and the free parameter δ of a quasiconformal kernel, are chosen with cross-validation method. (4) The number of projection vectors in each dimensionality reduction method is set to $M - 1$ in all experiments.

On the experimental procedure parameters selection, first select the optimal parameters of three Gaussian kernels and then the free parameter of the data-dependent kernel for Gaussian kernel, and followed by adaptive 2D Gaussian kernel. Performance evaluation: comparing the recognition performance of KPCA, 2DKPCA, adaptive Class-wise 2DKPCA. In our experiments, we implement our algorithm in the two-face databases, ORL face database and Yale face database. We select the following parameters for selection of Gaussian kernel and the free parameter for 2D Gaussian kernel, $\sigma^2 = 1 \times 10^5$, $\sigma^2 = 1 \times 10^6$, $\sigma^2 = 1 \times 10^7$ and $\sigma^2 = 1 \times 10^8$ for matrix-norm-based Gaussian kernel parameter, the free parameter of the data-dependent kernel, $\delta = 1 \times 10^5$, $\delta = 1 \times 10^6$, $\delta = 1 \times 10^7$, $\delta = 1 \times 10^8$, $\delta = 1 \times 10^9$, and $\delta = 1 \times 10^{10}$. Moreover, the dimension of the feature vector is set to 140 for ORL face database, and 70 for Yale face base. From the experimental results, we find that the higher recognition rate is obtained under the following parameters, $\sigma^2 = 1 \times 10^8$ and $\sigma^2 = 1 \times 10^5$ for ORL face database, and $\sigma^2 = 1 \times 10^8$ and $\delta = 1 \times 10^7$ for Yale face database. After we select the parameters for Adaptive 2D Gaussian kernel, we select the Gaussian parameter for Gaussian kernel and 2D Gaussian kernel. The same to the selection of parameters of Adaptive 2D Gaussian kernel, the $\sigma^2 = 1 \times 10^5$, $\sigma^2 = 1 \times 10^6$, $\sigma^2 = 1 \times 10^7$ and $\sigma^2 = 1 \times 10^8$ are selected to test the performance. From the experiments, we find that, on ORL face database, $\sigma^2 = 1 \times 10^6$ is selected for Gaussian, and $\sigma^2 = 1 \times 10^5$ is selected for 2D Gaussian kernel. And $\sigma^2 = 1 \times 10^8$ is selected for Gaussian, and $\sigma^2 = 1 \times 10^5$ is selected for 2D Gaussian kernel on Yale face database. All these parameters are used in the next section.

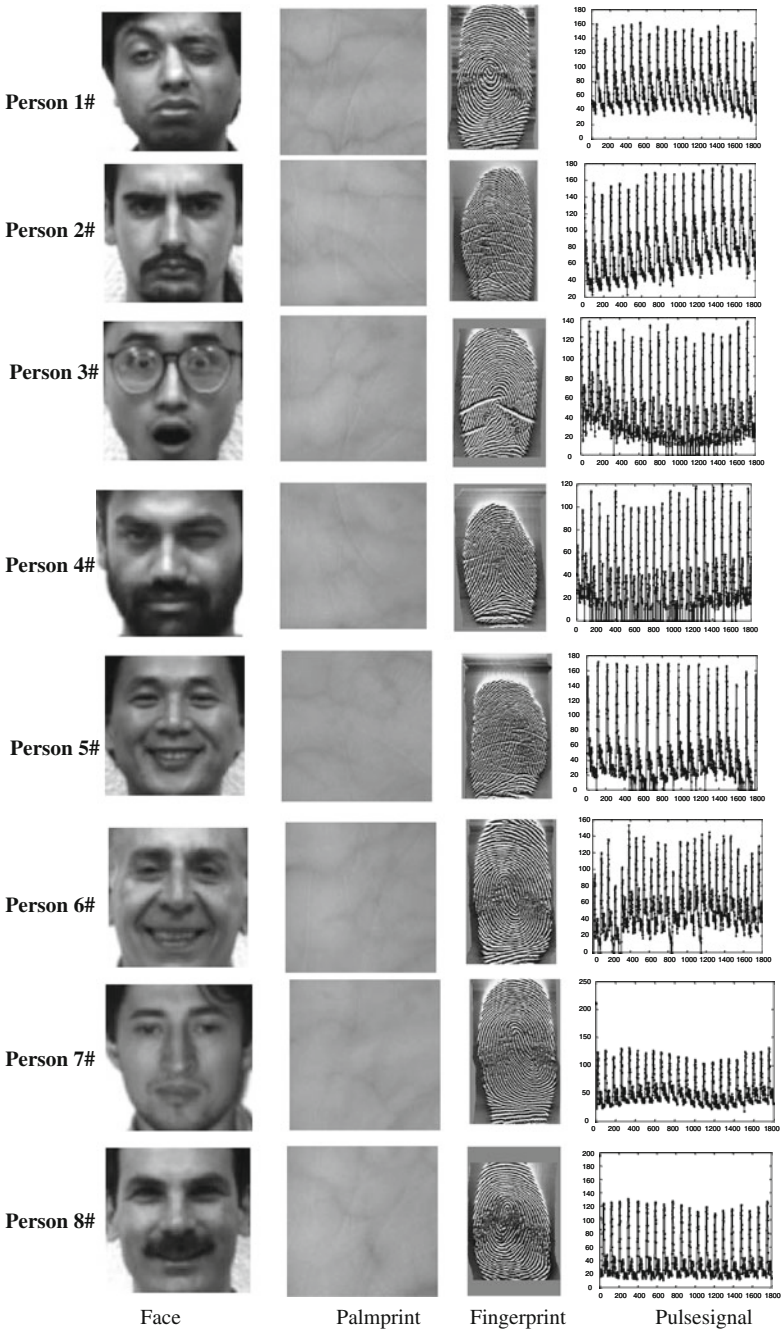


Fig. 8.3 Examples of input data

Table 8.1 Deterministic training and test set on YALE dataset

	SD1	SD2	SD3	SD4	SD5
Training set	1, 2, 3, 4, 5	10, 1, 2, 3, 4	8, 9, 10, 1, 2	7, 8, 9, 10, 1	6, 7, 8, 9, 10
Testing set	6, 7, 8, 9, 10	5, 6, 7, 8, 9,	3, 4, 5, 6, 7	2, 3, 4, 5, 6	1, 2, 3, 4, 5

Note 1 denotes the first sample of each person, 2 denotes the second sample of each person, and other samples are marked with the same ways

8.4.2 Results on Single Sensor Data

We implement the single sensor classification including face, palmprint, fingerprint, and wrist pulse signal. The proposed method is evaluation from the viewpoints of the efficiency and recognition performance. Principal component analysis (PCA), kernel discriminant analysis (KDA), fuzzy kernel Fisher discriminant (FKFD), locality preserving projection (LPP), class-wise locality preserving projection (CLPP), kernel class-wise locality preserving projection (KCLPP), and proposed adaptive class-wise 2DKPCA are implemented for the comparison.

On the face image sensor data applied the recognition accuracy to evaluate the performance of this single face image sensor. The experimental results are shown in Table 8.2. On the palmprint and fingerprint images, we implement the same experiments on the evaluations of the single-image sensory data. The experimental results are shown in Tables 8.3 and 8.4.

On the pulse signal sensory data, the analysis on the pulse signal based traditional pulse diagnosis including the pulse pattern reorganization, the arterial wave analysis [33, 34]. The pulse signal diagnosis has the three issues of data collection, feature extraction, and pattern classification. The pulse signals are collected using a Doppler ultrasound device and some preprocessing of the collected pulse signals has been performed. And the features with the characteristics of the measured pulse signals are extracted with the Doppler parameters [38, 39], Fourier transform [40], and wavelet transform [35, 41–44]. In these experiments, we apply the wavelet transform to feature extraction of wrist pulse signal for health evaluation. After extracting the features, the classification performance are evaluated with the popular classification methods including principal component analysis (PCA), kernel discriminant analysis (KDA), fuzzy kernel Fisher discriminant (FKFD), locality preserving projection (LPP), class-wise locality preserving projection

Table 8.2 Recognition performance on face images (%)

Datasets	PCA	KDA	FKFD	LPP	CLPP	KCLPP	Proposed
SD1	84.33	94.44	95.67	86.33	90.23	94.34	95.33
SD 2	86.77	92.22	93.33	90.67	91.11	94.32	95.33
SD 3	85.33	93.33	94.22	88.56	86.67	94.53	96.67
SD4	85.67	93.33	94.33	88.89	90.00	94.33	96.71
SD 5	88.67	96.67	97.67	95.56	93.33	97.67	98.87

Table 8.3 Recognition performance on fingerprint images (%)

Datasets	PCA	KDA	FKFD	LPP	CLPP	KCLPP	Proposed
SD1	85.36	95.54	96.35	87.37	91.44	95.23	96.35
SD 2	87.65	93.34	94.34	91.77	92.25	95.35	96.35
SD 3	86.45	94.54	95.45	89.65	87.77	95.76	97.55
SD4	86.45	94.53	95.54	88.98	91.23	95.13	97.54
SD 5	89.34	97.45	98.13	96.45	94.35	98.12	99.21

Table 8.4 Recognition performance on palmprint images (%)

Datasets	PCA	KDA	FKFD	LPP	CLPP	KCLPP	Proposed
SD1	83.23	93.25	94.56	85.14	89.24	93.32	94.16
SD 2	85.67	91.24	92.45	89.24	90.12	93.33	94.14
SD 3	84.33	92.45	93.15	87.25	85.36	93.23	95.37
SD4	84.23	92.45	93.13	87.65	89.24	93.12	95.24
SD 5	87.77	95.24	96.17	94.24	92.35	96.33	97.25

Table 8.5 Recognition performance on wrist pulse signal (%)

Datasets	PCA	KDA	FKFD	LPP	CLPP	KCLPP	Proposed
SD1	85.25	91.25	92.15	87.15	88.25	92.25	93.15
SD 2	86.85	89.25	90.45	85.25	87.25	91.15	93.15
SD 3	85.50	90.25	91.15	86.15	87.25	91.05	94.25
SD4	85.75	90.25	91.25	86.45	87.25	91.15	94.55
SD 5	88.75	92.25	92.15	87.25	88.25	92.25	95.55

(CLPP), kernel class-wise locality preserving projection (KCLPP), and the proposed adaptive Class-wise 2DKPCA. The experimental results are shown in Table 8.5.

8.4.3 Results on Multisensor Data

We implement the multisensor data for identification and health analysis. The classification performances are evaluating on the classification methods including principal component analysis (PCA), kernel discriminant analysis (KDA), fuzzy kernel Fisher discriminant (FKFD), locality preserving projection (LPP), class-wise locality preserving projection (CLPP), kernel class-wise locality preserving projection (KCLPP), and the proposed adaptive class-wise KPCA. Table 8.6 describes the results of the identification and health analysis based on different classification methods. In these experiments, we test the algorithms on all training samples.

Table 8.6 Recognition performance on wrist pulse signal (%)

	PCA	KDA	FKFD	LPP	CLPP	KCLPP	Proposed
Identification	89.15	96.25	97.15	95.25	93.35	97.10	98.15
Health	87.15	89.55	90.15	86.25	88.15	90.25	93.15

References

1. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Netw 12:181–201
2. Li J-B, Chu S-C, Pan J-S, Ho J-H (2007) Adaptive data-dependent matrix norm based Gaussian kernel for facial feature extraction. Int J Innovative Comput, Inf Control 3(5):1263–1272
3. Sahbi H (2007) Kernel PCA for similarity invariant shape recognition. Neurocomputing 70:3034–3045
4. Mika S, Ratsch G, Weston J, Schölkopf B, Muller K-R (1999) Fisher discriminant analysis with kernels. In: Proceedings of IEEE international workshop neural networks for signal processing IX, pp 41–48
5. Tao D, Tang X, Li X, Wu X (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Trans Pattern Anal Mach Intell 28(7):1088–1099
6. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Netw 14(1):117–226
7. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
8. Liang Z, Shi P (2005) Uncorrelated discriminant vectors using a kernel method. Pattern Recogn 38:307–310
9. Liang Z, Shi P (2004) Efficient algorithm for kernel discriminant anlaysis. Pattern Recogn 37(2):381–384
10. Liang Z, Shi P (2004) An efficient and effective method to solve kernel Fisher discriminant analysis. Neurocomputing 61:485–493
11. Yang MH (2002) Kernel eigenfaces versus kernel fisherfaces: face recognition using kernel methods. In: Proceedings of fifth ieee international conference automatic face and gesture recognition, pp 215–220
12. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Netw 14(1):117–126
13. Zheng W, Zou C, Zhao L (2005) Weighted maximum margin discriminant analysis with kernels. Neurocomputing 67:357–362
14. Huang J, Yuen PC, Chen W-S, Lai JH (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. In: Proceedings of the sixth IEEE international conference on automatic face and gesture recognition
15. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst, Man and Cybern-Part B: Cybern 35(3):556–562
16. Chen W-S, Yuen PC, Huang J, Dai D-Q (2005) Kernel machine-based one-parameter regularized fisher discriminant method for face recognition. IEEE Trans Syst, Man and Cybern-Part B: Cybern 35(4):658–669
17. Liang Y, Li C, Gong W, Pan Y (2007) Uncorrelated linear discriminant analysis based on weighted pairwise Fisher criterion. Pattern Recogn 40:3606–3615
18. Zheng Y-J, Yang J, Yang J-Y, Wu X-J (2006) A reformative kernel Fisher discriminant algorithm and its application to face recognition. Neurocomputing 69(13-15):1806–1810
19. Tao D, Tang X, Li X, Rui Y (2006) Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. IEEE Trans Multimedia 8(4):716–727

20. Xu Y, Zhang D, Jin Z, Li M, Yang J-Y (2006) A fast kernel-based nonlinear discriminant analysis for multi-class problems. Pattern Recogn 39(6):1026–1033
21. Saadi K, Talbot NLC, Cawley C (2007) Optimally regularised kernel Fisher discriminant classification. Neural Netw 20(7):832–841
22. Yeung D-Y, Chang H, Dai G (2007) Learning the kernel matrix by maximizing a KFD-based class separability criterion. Pattern Recogn 40(7):2021–2028
23. Shen LL, Bai L, Fairhurst M (2007) Gabor wavelets and general discriminant analysis for face identification and verification. Image Vis Comput 25(5):553–563
24. Ma B, Qu H-Y, Wong H-S (2007) Kernel clustering-based discriminant analysis. Pattern Recogn 40(1):324–327
25. Wu X-H, Zhou J-J (2006) Fuzzy discriminant analysis with kernel methods. Pattern Recogn 39(11):2236–2239
26. Liu Q, Lu H, Ma S (2004) Improving kernel Fisher discriminant analysis for face recognition. IEEE Trans Pattern Analysis Mach Intell 14(1):42–49
27. Shu J, Sun Y (2007) Developing classification indices for Chinese pulse diagnosis. Complement Ther Med 15 (3):190–198
28. Leonard P, Beattie T, Addison P, Watson J (2004) Wavelet analysis of pulse oximeter waveform permits identification of unwell children. Emerg Med J 21:59–60
29. Zhang Y, Wang Y, Wang W, Yu J (2002) Wavelet feature extraction and classification of Doppler ultrasound blood flow signals. J Biomed Eng 19(2):244–246
30. Lu W, Wang Y, Wang W (1999) Pulse analysis of patients with severe liver problems. IEEE Eng Med Biol Mag 18 (1):73–75
31. Zhang A, Yang F (2005) Study on recognition of sub-health from pulse signal. In: Proceedings of the ICNNB conference, 3:1516–1518
32. Zhang D, Zhang L, Zhang D, Zheng Y (2008) Wavelet-based analysis of Doppler ultrasonic wrist-pulse signals. In: Proceedings of the ICBBE conference, Shanghai 2:539–543
33. Chen Y, Zhang L, Zhang D, Zhang D (2009) Wrist pulse signal diagnosis using modified Gaussian models and fuzzy C-means classification. Med Eng Phys 31:1283–1289
34. Chen Y, Zhang L, Zhang D, Zhang D (2011) Computerized wrist pulse signal diagnosis using modified auto-regressive models. J Med Syst 35:321–328
35. Chen B, Wang X, Yang S, McGreavy C (1999) Application of wavelets and neural networks to diagnostic system development, 1, feature extraction. Comput Chem Eng 23:899–906
36. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces versus fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Analysis Mach Intell 19(7):711–720
37. Samaria F, Harter A (1994) Parameterisation of a stochastic model for human face identification. In: Proceedings of 2nd IEEE workshop on applications of computer vision, pp 138–142
38. Wang Y,WuX, Liu B, Yi Y (1997) Definition and application of indices in Doppler ultrasound sonogram. J Biomed Eng Shanghai, 18:26–29
39. Ruano M, Fish P (1993) Cost/benefit criterion for selection of pulsed Doppler ultrasound spectral mean frequency and bandwidth estimators. IEEE Trans BME 40:1338–1341
40. Lu W, Wang Y, Wang W (1999) Pulse analysis of patients with severe liver problems. IEEE Eng Med Biol Mag 18(Jan/Feb (1)): 73–75
41. Leonard P, Beattie TF, Addison PS, Watson JN (2004) Wavelet analysis of pulse oximeter waveform permits identification of unwell children. J Emerg Med 21:59–60
42. Zhang Y, Wang Y, Wang W, Yu J (2002) Wavelet feature extraction and classification of Doppler ultrasound blood flow signals. J Biomed Eng 19(2):244–246
43. Zhang D, Zhang L, Zhang D, Zheng Y (2008) Wavelet based analysis of Doppler ultrasonic wrist-pulse signals. In: Proceedings of the ICBBE 2008 conference, vol. 2, pp 539–543
44. Hera1 A, Hou Z (2004) Application of wavelet approach for ASCE structural health monitoring benchmark studies. J Eng Mech, 1:96–104

Chapter 9
Kernel-Optimization-Based Face Recognition

9.1 Introduction

Feature extraction is an important step and essential process in many data analysis areas, such as face recognition, handwriting recognition, human facial expression analysis, speech recognition. As the most popular method for feature extraction, dimensionality reduction (DR) has been widely studied and many DR methods were proposed such as linear discriminant analysis (LDA) and principal component analysis (PCA) [1, 2]. These methods are linear dimensionality reduction and fail to capture a nonlinear relationship of the data. Kernel method was widely used to DR owing to its excellent performance on discovering the complicated nonlinear relationships of input data [3]. Accordingly, kernel version of linear DR methods was developed, such as kernel principal component analysis (KPCA), kernel discriminant analysis (KDA) [4], and its improved methods (Baudat and Anouar [5], Liang and Shi [6], Wang [7], Chen [8] and Lu [9]). Moreover, other kernel methods, such as kernel locality preserving projection (KLPP), have been widely studied and used in many areas such as face recognition and radar target recognition [10, 11], and other researchers also improved LPP with kernels in the previous works [12, 13, 14, 4]. Kernel learning methods improve the performances of many linear feature extraction methods owing to the self-adaptive data distributions for classification. Kernel functions have a heavy influence on kernel learning, because the geometrical structure of the data in the kernel mapping space is totally determined by the kernel function. The discriminative ability of the data in the feature space could be even worse with an inappropriate kernel. The method, optimizing kernel parameters from a set of discrete values, was widely studied [13, 15, 16], but this method did not change the geometry structure of the data for classification. Xiong proposed a data-dependent kernel for kernel optimization for the different [17], and Amari presented support vector machine classifier through modifying the kernel function [1]. In our previous works [15, 18], we present data-dependent kernel-based KDA algorithm for face recognition application.

As above discussions, kernel learning is an important research topic in the machine learning area, and some theory and application fruits are achieved and

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_9,

widely applied in pattern recognition, data mining, computer vision, and image and signal processing areas. The nonlinear problems are solved with kernel function, and system performances such as recognition accuracy, prediction accuracy are largely increased. However, kernel learning method still endures a key problem, i.e., kernel function and its parameter selection. Kernel function and its parameters have the direct influence on the data distribution in the nonlinear feature space, and the inappropriate selection will influence the performance of kernel learning. In this book, we focus on two schemes: One is kernel optimization algorithm and procedure, and the other is the framework of kernel learning algorithms. To verify the effectiveness of the kernel optimization scheme proposed, the proposed kernel optimization method is applied into popular kernel learning methods, including kernel principal component analysis, kernel discriminant analysis, and kernel locality preserving projection.

Kernel learning has become an important research topic of machine learning area, and it has wide applications in pattern recognition, computer vision, and image and signal processing. Kernel learning provides a promising solution to the nonlinear problems including nonlinear feature extraction, classification, and clustering. However, kernel-based system still endures the problem of how to select the kernel function and its parameters. Previous researches presented the method of choosing the parameters from a discrete value set, and this method did not change the structure of data distribution in kernel-based mapping space. Accordingly, the performance did not increase owing to the unchangeable data distribution. Based on this motivation, we present a uniform framework of kernel self-optimization with the ability of adjusting data structure. In this framework, firstly data-dependent kernel is extended for the higher ability of kernel structure adjusting, and secondly, two criterions of measuring the data discrimination are used to solve the optimal parameter. Some evaluations are implemented to testify the performance on popular kernel learning methods including kernel principal component analysis (KPCA), kernel discriminant analysis (KDA), and kernel locality preserving projection (KLPP). These evaluations show that the framework of kernel self-optimization is feasible to enhance kernel-based learning methods.

9.2 Data-Dependent Kernel Self-optimization

9.2.1 Motivation and Framework

Kernel matrix computing is the key step of kernel-based learning for the classification, clustering, and other statistical pattern analysis. This procedure causes two problems: computation burden and kernel selection. In this book, we aim to present a framework of kernel optimization to improve the kernel learning performance. Instead of only choosing the parameters for kernel from a set of discrete values of the parameters, we apply data-dependent kernel to kernel optimization

because the geometrical structures of the data in the kernel mapping space are adaptively changeable through adjusting the parameter of data-dependent kernel. This contribution of this book is to present a uniform framework of kernel optimization to solve the kernel selection problem widely endured by kernel-based learning method in the practical applications. Although some improved kernel learning methods based on adaptive kernel selection have been proposed in author's previous work, a general framework of kernel optimization has not been proposed in the previous work. So, different from author's previous work [18], this book presents a uniform framework of kernel optimization for kernel-based learning. The framework of kernel optimization is applied into three kinds of kernel-based learning methods to demonstrate the feasibility of kernel optimization. Figure 9.1 shows the framework of kernel optimization with its application. The framework includes kernel optimizing, training and testing for kernel-based feature extraction, and recognition.

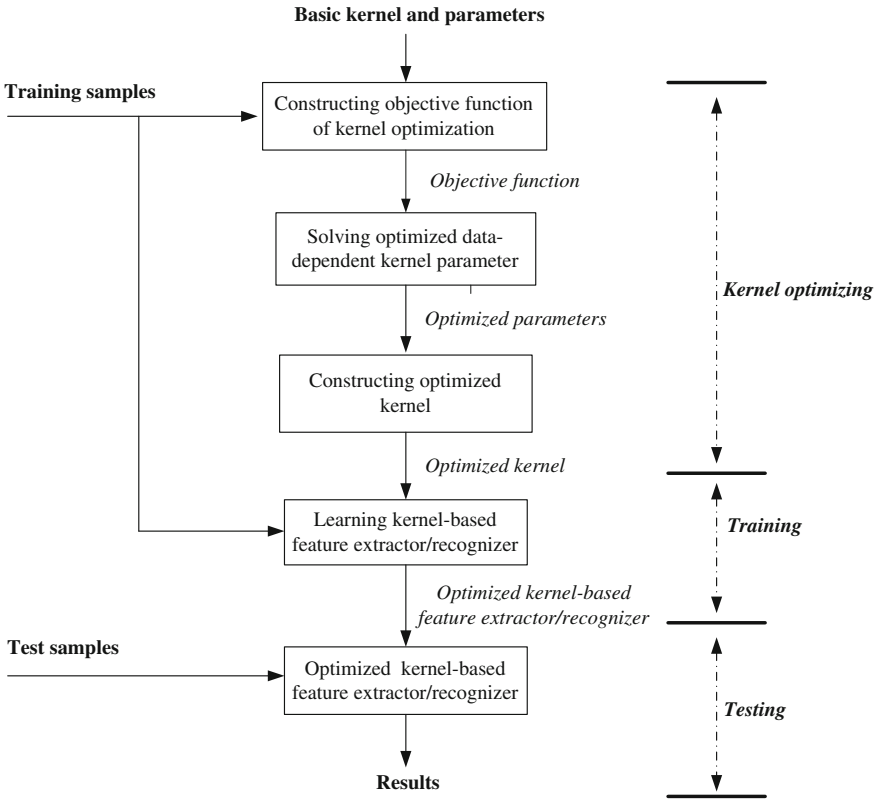


Fig. 9.1 Framework of kernel optimization and its application

9.2.2 Extended Data-Dependent Kernel

The recognition accuracy is improved with the extended data-dependent kernel with the adaptive parameter through constructing the constraint optimization equation. Data-dependent kernel is defined as [15]

$$k(x, y) = f(x)f(y)k_0(x, y), \tag{9.1}$$

where $k_0(x, y)$ is the basic kernel, for example, polynomial kernel and Gaussian kernel. $f(x)$ is the positive value of x as follows:

$$f(x) = a_0 + \sum_{n=1}^{N_{EV}} a_n e(x, z_n), \tag{9.2}$$

where z_n is the nth expansion vector (EV), N_{EV} is the number of support vectors, and a_n represents the contribution of z_n, $e(x, z_n)$ represents the similarity between z_n and x. We propose four versions of defining $e(x, z_n)$ marked with M1, M2, M3, and M4, respectively, for different applications as follows:

M1, $e(x, z_n) = \begin{cases} 1 & x, z_n : \textit{same class} \\ e^{-\delta\|x-z_n\|^2} & x, z_n : \textit{different class} \end{cases}$, regards all the class-labeled training samples as the expansion vectors and centralizes the same class of samples into one point in the feature space.

M2, $e(x, z_n) = e^{-\delta\|x-z_n\|^2}$, considers all training samples as the expansion vectors as in the traditional method. The number of training samples is equal to the number of expansion vectors.

M3, $e(x, z_n) = \begin{cases} 1 & x, \overline{z_n} : \textit{same class} \\ e^{-\delta\|x-\overline{z_n}\|^2} & x, \overline{z_n} : \textit{different class} \end{cases}$, considers the mean vectors of all samples as expansion vectors, where $\overline{z_n}$ is the mean vector of all samples. The small number of expansion vectors decreases the computation stress.

M4, $e(x, z_n) = e^{-\delta\|x-\overline{z_n}\|^2}$, regards the mean vector of samples as the expansion vector. This method only considers the distance between any sample and the center of all samples without considering the class label of each sample.

According to the extended definition of data-dependent kernel function, suppose the free parameter δ and expansion vector z_n, $n = 1, 2, \ldots, N_{EV}$, the geometry structure of the data is changed through adjusting the expansion coefficient a_n. So, we construct the constraint optimization equation of maximizing discrimination to find the optimal expansion coefficients.

9.2.3 Kernel Optimization

We present two kernel optimization methods based on Fisher criterion and maximum margin criterion. Under the different expansion coefficient vector a_n, the

geometry structure of data in the empirical space causes the discriminative ability of samples. Let $A = [a_1, a_2, \ldots, a_n]^T$, where $A^T A = 1$, then we obtain the following optimization equation:

$$\max Q(A),$$
$$s.t. \quad A^T A - 1 = 0. \tag{9.3}$$

The definition of $Q(A)$-based Fisher criterion and maximum margin criterion is shown as follows:

$$Q(A) = \begin{cases} \frac{A^T E^T B_0 E A}{A^T E^T W_0 E A}, & FC \\ A^T (2S_B - S_T) A, & MMC \end{cases}, \tag{9.4}$$

where FC represents Fisher criterion and MMC represents maximum margin criterion.$E = \begin{bmatrix} 1 & e(x_1, z_1) & \ldots & e(x_1, z_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e(x_n, z_1) & \ldots & e(x_n, z_n) \end{bmatrix}$ is the matrix of using $e(x, z_n)$ with different methods, M1, M2, M3, and M4 for different applications. In the following experiments, we evaluate M1, M2, M3, and M4 methods of defining $e(x, z_n)$ in the different databases. For c classes of training sample sets, the matrix, K_{ii}, $i = 1, 2, \ldots, c$, denotes the kernel matrix of the cth class of n_i training samples, and K is the kernel matrix of all n training samples. K_{ii}, $i = 1, 2, \ldots, c$, is the element of the kernel matrix. Then, the matrices B_0 and W_0 in (9.4) are computed with

$$B_0 = \begin{bmatrix} \frac{1}{n_1} K_{11} & & & \\ & \frac{1}{n_2} K_{22} & & \\ & & \ddots & \\ & & & \frac{1}{n_c} K_{cc} \end{bmatrix} - \frac{1}{n} K, \tag{9.5}$$

and

$$W_0 = \begin{bmatrix} k_{11} & & & \\ & k_{22} & & \\ & & \ddots & \\ & & & k_{nn} \end{bmatrix} - \begin{bmatrix} \frac{1}{n_1} K_{11} & & & \\ & \frac{1}{n_2} K_{22} & & \\ & & \ddots & \\ & & & \frac{1}{n_c} K_{cc} \end{bmatrix}, \tag{9.6}$$

S_T is the total scatter matrix, $S_T = \left(Y - \frac{1}{n} Y 1_n^T 1_n\right)\left(Y - \frac{1}{n} Y 1_n^T 1_n\right)^T$, where $Y = KP\Lambda^{-\frac{1}{2}}$. S_B is the between-class scatter matrix, and $u_1, u_2, \ldots, u_c$ are the mean vectors of each classes, and $1_c = [1, 1, \ldots, 1]_{1 \times c}$. Then, $S_B = \left(\sqrt{n_1}(u_1 - u), \ldots, \sqrt{n_c}(u_c - u)\right)\left(\sqrt{n_1}(u_1 - u), \ldots, \sqrt{n_c}(u_c - u)\right)^T$.

The solution of Eq. (9.4) is shown in Appendix.

The initialized learning rate ε_0 and the total iteration number N are chosen in advance. The initial learning rate ε_0 influences the convergence speed of the algorithm, and the total iteration number N determines the time of solution. Only when the parameters ε_0 and N are chosen appropriately, then the optimal expansion coefficient vector is solved. So the solution of expansion coefficient is not unique, which is determined by the selection of learning parameter. The iteration algorithm costs much time. Fisher-criterion-based kernel optimization method solves the optimal solution with the iteration method, while maximum margin criterion method uses the eigenvalue equation. Fisher criterion method costs much more time than maximum margin criterion method. Moreover, Fisher criterion method needs to choose the procedural parameters.

The geometry structure of sample data in the nonlinear projection space is different with the different kernel function. Accordingly, data in the nonlinear projection space have the different class discriminative ability. So the kernel function should be dependent on the input data, which is the main idea of data-dependent kernel which was proposed in [19]. The parameter of the data-dependent kernel is changed according to the input data so that the optimal geometry structure of data in the feature space is achieved for the classification. In this book, we extend the definition of the data-dependent kernel as the objective function for creating the constrained optimization equation to solve the solution.

$$k(x, y) = f(x)f(y)k_0(x, y) \tag{9.7}$$

where $k_0(x,y)$ is the basic kernel function, such as polynomial kernel and Gaussian kernel. The function $f(x)$ is defined as [19]

$$f(x) = \sum\nolimits_{i \in SV} a_i e^{-\delta \|x - \tilde{x}_i\|^2} \tag{9.8}$$

where $\tilde{x}_i$ is the support vector, SV is the set of support vector, a_i denotes the positive value which represents the distribution of $\tilde{x}_i$, δ is the free parameter. We extend the definition of data-dependent kernel through defining the function $f(x)$ with the different ways as follows.

$$f(x) = b_0 + \sum_{n=1}^{N_{XV}} b_n e(x, \tilde{x}_n) \tag{9.9}$$

where δ is the free parameters, $\tilde{x}_i$ is the expansion vectors (xvs), N_{XV} is the number of expansion vectors, and b_n ($n = 0, 1, 2, \ldots, N_{XV}$) is the according expansion coefficients. Xiong [20] selected randomly the one-third of total number of samples as the expansion vectors. This book proposed four methods of defining $e(x, \tilde{x}_n)$ as follows:

m1:

$$\begin{aligned} e(x, \tilde{x}_n) &= e(x, x_n) \\ &= \begin{cases} 1 & x \text{ and } x_n \text{ with the same class label information} \\ e^{-\delta \|x - x_n\|^2} & x \text{ and } x_n \text{ with the different class label information} \end{cases} \end{aligned}$$

This method regards the all labeled samples as the expansion vector. This method is applicable to supervised learning method. This method must consider the class label of samples and causes the samples from the same class labels centralize into one point in the feature space. Currently, many pattern recognition methods are supervised learning methods, such as linear discriminant analysis, support vector machine.

m2:

$$e(x, \tilde{x}_n) = e(x, x_n) = e^{-\delta\|x-x_n\|^2}, \quad (n = 1, 2, \ldots, M)$$

All training samples are considered as the expansion vectors. The total number of expansion vectors is equal to the total number of all training samples, i.e., $M = \text{XVs}$.

m3:

$$\begin{aligned} e(x, \tilde{x}_n) &= e(x, x_n) \\ &= \begin{cases} 1 & \text{The class label information of } x \text{ and } \bar{x} \text{ is same;} \\ e^{-\delta\|x-\bar{x}_n\|^2} & \text{The class label information of } x \text{ and } \bar{x} \text{ is different;} \end{cases} \end{aligned}$$

where $N_{XV} = L$, $\bar{x}_n$ is the class mean of the *nth* class of samples. This method is to solve the computation problem faced by methods *m1* and *m2*.

m4:

$$e(x, \tilde{x}_n) = e(x, \bar{x}_n) = e^{-\delta\|x-\bar{x}_n\|^2}, \quad (n = 1, 2, \ldots, L)$$

This method considers the class mean as the expansion vector. This method considers the distance between any samples with the mean of sample but rarely considers the class label.

According to the extended definition of data-dependent kernel function, suppose the free parameter δ and expansion vector $\tilde{x}_n (n = 0, 1, 2, \ldots, N_{XV})$, the geometry structure of the data in the nonlinear mapping projection space is changeable with the changing of the expansion coefficient $\alpha_n (n = 0, 1, 2, \ldots, N_{XV_s})$. So we can adjust the geometry structure of data in the nonlinear mapping space through changing the expansion coefficient.

In order to optimize the kernel function through finding the optimal expansion coefficient, consider the computation problem, we optimize the kernel function in the empirical feature space [20].

Suppose that $\{x_i\}_{i=1}^n$ be d-dimensional training samples. X is the $n \times d$ matrix with its column x_i^T, $i = 1, 2, \ldots, n$, K is $n \times n$ kernel matrix $K = \left[K_{ij}\right]_{n\times n}$. The element of matrix k_{ij} is

$$k_{ij} = \Phi(x_i) \times \Phi\left(x_j\right) = k\left(x_i, x_j\right) \tag{9.10}$$

where K is positive definite symmetrical matrix. So the matrix is decomposed into

$$K_{n\times n} = P_{n\times r} \wedge_{r\times r} P_{r\times n}^T \tag{9.11}$$

where $\wedge$ is the diagonal matrix consisted of r positive eigenvalue of kernel matrix. P is the matrix consisted of eigenvectors corresponding to the eigenvalue. Then, the mapping from the input space to r dimensional space is defined as

$$\Phi_r^e : \chi \to R^r$$

$$x \to \wedge^{-1/2} P^T (k(x, x_1), k(x, x_2), \ldots, k(x, x_n))^T \tag{9.12}$$

This mapping is called empirical kernel map. Accordingly, the mapping space $\Phi_r^e(\chi) \subset R^r$ is called empirical feature space.

1. **Fisher-criterion-based kernel optimization**

Fisher criterion is used to measure the class discriminative ability of the samples in the empirical feature space [20]. The discriminative ability of samples in the empirical feature space is defined as

$$J_{\text{Fisher}} = \frac{tr(S_B^\Phi)}{tr(S_W^\Phi)} \tag{9.13}$$

where J_{Fisher} measure the linear discriminative ability, S_B^Φ is the between-class scatter matrix, S_W^Φ is interclass scatter matrix, and tr denotes the trace. Let K is the kernel matrix with its element k_{ij} $(i, j = 1, 2, \ldots, n)$ is calculated with x_i and x_j. The matrix K_{pq}, $p, q = 1, 2, \ldots, L$ is the $n_p \times n_q$ matrix with p and q classes. Then, in the empirical feature space, we can obtain $tr(S_B^\Phi) = 1_n^T B 1_n$ and $tr(S_W^\Phi) = 1_n^T W 1_n$, where $B = \text{diag}\left(\frac{1}{n_1} K_{11}, \frac{1}{n_2} K_{22}, \ldots, \frac{1}{n_L} K_{LL}\right) - \frac{1}{n} K$. The class discriminative ability is defined as

$$J_{\text{Fisher}} = \frac{1_n^T B 1_n}{1_n^T W 1_n} \tag{9.14}$$

According to the definition of the data-dependent kernel, let $D = \text{diag}(f(x_1), f(x_2), \ldots, f(x_n))$ the relation between the data-dependent kernel matrix K and the basic kernel matrix K_0 calculated with basic kernel function $K_0(x, y)$ is defined as

$$K = DK_0D \tag{9.15}$$

Accordingly, $B = DB_0D$ and $W = DW_0D$. Then,

$$J_{\text{Fisher}} = \frac{1_n^T\, DB_0\, D1_n}{1_n^T\, DW_0\, D1_n} \tag{9.16}$$

where 1_n is n-dimensional unit vector, according to the definition of data-dependent kernel, then

$$D1_n = E\alpha \tag{9.17}$$

where, $\alpha = [a_0, a_1, a_2, \ldots, a_{N_{\text{XVs}}}]^T$, the matrix E is

$$E = \begin{bmatrix} 1 & e(x_1, \tilde{x}_1) & \ldots & e(x_1, \tilde{x}_{N_{\text{XVs}}}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e(x_n, \tilde{x}_1) & \ldots & e(x_n, \tilde{x}_{N_{\text{XVs}}}) \end{bmatrix}$$

Then,

$$J_{\text{Fisher}} = \frac{a^T E^T B_0 E \alpha}{a^T E^T W_0 E \alpha} \tag{9.18}$$

where $E^T B_0 E$ and $E^T W_0 E$ are constant matrix and J_{Fisher} is a function with its variable α. Under the different expansion coefficient vector α, the geometry structure of data in the empirical space causes the discriminative ability of samples. Our goal is to find the optimal α to maximize J_{Fisher}. Suppose that α is a unit vector, i.e., $\alpha^T \alpha = 1$. ,the constrained equation is created to solve the optimal α as follows:

$$\begin{aligned} &\max \quad J_{\text{Fisher}}(\alpha) \\ &\text{subject to} \quad \alpha^T \alpha - 1 = 0 \end{aligned} \tag{9.19}$$

There are many methods of solving the above optimization equation. The following method is a classic method. Let $J_1(\alpha) = \alpha^T E^T B_0 E \alpha$ and $J_2(\alpha) = \alpha^T E^T W_0 E \alpha$, then

$$\frac{\partial J_1(\alpha)}{\alpha} = 2E^T B_0 E \alpha \tag{9.20}$$

$$\frac{\partial J_2(\alpha)}{\alpha} = 2E^T W_0 E \alpha \tag{9.21}$$

then

$$\frac{\partial J_{\text{Fisher}}(\alpha)}{\partial \alpha} = \frac{2}{J_2^2} \left(J_2 E^T B_0 E - J_1 E^T W_0 E \right) \alpha \tag{9.22}$$

In order to maximize J_{Fisher}, let $\partial J_{\text{Fisher}}(\alpha)/\partial \alpha = 0$, then

$$J_1 E^T W_0 E \alpha = J_2 E^T B_0 E \alpha \tag{9.23}$$

If $(E^T W_0 E)^{-1}$ exists, then

$$J_{\text{Fisher}} \alpha = \left(E^T W_0 E \right)^{-1} \left(E^T B_0 E \right) \alpha \tag{9.24}$$

J_{Fisher} is equal to the eigenvalue of $(E^T W_0 E)^{-1}(E^T B_0 E)$, and the corresponding eigenvector is equal to expansion coefficients vector α. In many applications, the

matrix $(E^T W_0 E)^{-1}(E^T B_0 E)$ is not symmetrical or $E^T WE$ is singular. So the iteration method is used to solve the optimal α, that is,

$$\alpha^{(n+1)} = \alpha^{(n)} + \varepsilon\left(\frac{1}{J_2}E^T B_0 E - \frac{J_{\text{Fisher}}}{J_2}E^T W_0 E\right)\alpha^{(n)} \tag{9.25}$$

ε is the learning rate as follows. The definition of learning rate is

$$\varepsilon(n) = \varepsilon_0\left(1 - \frac{n}{N}\right) \tag{9.26}$$

where ε_0 is the initialized learning rate, n and N are the current iteration number and the total iteration number in advance, respectively.

The initialized learning rate ε_0 and the total iteration number N are set in advance for the solution of the expansion coefficient. The initial learning rate ε_0 influences the convergence speed of the algorithm, and the total iteration number N determines the time of solution. Only when the parameters ε_0 and N are chosen appropriately, we choose the optimal expansion coefficient vector. So the solution of expansion coefficient is not unique, which is determined by the selection of learning parameter. The iteration algorithm costs much time. So we select the maximum margin criterion as the objective function to solve the optimal expansion coefficients.

2. **Maximum margin criterion (MMC)-based kernel optimization**

Kernel optimization is also implemented in the empirical feature space. Compared with Fisher-criterion-based kernel optimization, maximum-margin-criterion-based kernel optimization uses MMC to create the constrained optimization equation to solve the optimal expansion coefficient vector. MMC is used for feature extraction [1], and the main idea is to maximize the margin of the different class of samples. The average margin the class c_i and class c_j is obtained as follows:

$$\text{Dis} = \frac{1}{2n}\sum_{i=1}^{L}\sum_{j=1}^{L} n_i n_j d(c_i, c_j) \tag{9.27}$$

where $d(c_i, c_j) = d\left(m_i^{\Phi}, m_j^{\Phi}\right) - S(c_i) - S(c_j)$ denotes the margin between class i and class j, $d\left(m_i^{\Phi}, m_j^{\Phi}\right)$ denotes the distance between the centers of two classes of samples, and $S(c_i)$ denotes the scatter matrix $c_i (i = 1, 2, \ldots, L)$ which is defined as follows:

$$S(c_i) = tr\left(S_i^{\Phi}\right) \tag{9.28}$$

where $tr\left(S_i^{\Phi}\right) = \frac{1}{n_i}\sum_{p=1}^{n_i}\left(\Phi(x_i^p) - m_i^{\Phi}\right)^T\left(\Phi(x_i^p) - m_i^{\Phi}\right)$,$S_i^{\Phi}$ is the scatter matrix of ith class.

Suppose that $tr\left(S_B^{\Phi}\right)$ and $tr\left(S_W^{\Phi}\right)$ are the trace of the between-class and interclass scatter matrix, then $\text{Dis} = tr\left(S_B^{\Phi}\right) - tr\left(S_W^{\Phi}\right)\circ$

It is easy to obtain

$$\text{Dis} = tr\left(2S_B^{\Phi} - S_T^{\Phi}\right) \tag{9.29}$$

In the empirical feature space, the sample set $Y = KP\wedge^{-1/2}$, where K is the data-dependent kernel. P and $\wedge$ satisfies

$$K = P\wedge^T P^T \tag{9.30}$$

Rewrite S^T as

$$S^T = \left(Y - \frac{1}{m}Y1_m^T 1_m\right)\left(Y - \frac{1}{m}Y1_m^T 1_m\right)^T \tag{9.31}$$

where $1_m = [1, 1, \ldots, 1]_{1\times m}$, then

$$\text{trace}(S_T) = 1_m\left(Y - \frac{1}{m}Y1_m^T 1_m\right)^T\left(Y - \frac{1}{m}Y1_m^T 1_m\right)1_m^T \tag{9.32}$$

Since $D1_n = E\alpha$, then

$$\begin{aligned}\text{trace}(S_T) &= 1_m D\left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)^T\left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)D1_m^T \\ &= (E\alpha)^T\left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)^T\left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)E\alpha \\ &= \alpha^T\left(\left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)E\right)^T\left(\left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)E\right)\alpha\end{aligned} \tag{9.33}$$

Let $X_T = \left(Y_0 - \frac{1}{m}Y_0 1_m^T 1_m\right)E$, then

$$\text{trace}(S_T) = \alpha^T (X_T)^T X_T \alpha \tag{9.34}$$

where $Y_0 = K_0 P_0 \wedge_0^{-1/2}$, $K_0 = P_0 \wedge_0^T P_0^T$ and K_0 is the basic kernel matrix. Similarly, it easy to obtain

$$S_B = (\sqrt{m_1}(u_1 - u), \ldots, \sqrt{m_c}(u_c - u))(\sqrt{m_1}(u_1 - u), \ldots, \sqrt{m_c}(u_c - u))^T \tag{9.35}$$

where $1_c = [1, 1, \ldots, 1]_{1\times c}$. Then, $\text{trace}(S_B)$ is defined as

$$\text{trace}(S_B) = 1_m M Y^T Y M^T 1_m^T = 1_m\left(YM^T\right)^T\left(YM^T\right)1_m^T \tag{9.36}$$

Let $M = M_1 - M_2$ and

$$M_1 = \begin{bmatrix} \left[\frac{1}{\sqrt{m_1}}\right]_{m_1\times m_2} & 0_{m_1\times m_2} & \cdots & 0_{m_1\times m_c} \\ 0_{m_2\times m_1} & \left[\frac{1}{\sqrt{m_2}}\right]_{m_2\times m_2} & \cdots & 0_{m_2\times m_c} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{m_c\times m_1} & 0_{m_c\times m_2} & \cdots & \left[\frac{1}{\sqrt{m_c}}\right]_{m_c\times m_c} \end{bmatrix} \text{ and}$$

$$\mathrm{M}_2 = \begin{bmatrix} \frac{\sum_j^c \sqrt{m_j}}{m} & 0 & \cdots & 0 \\ 0 & \frac{\sum_j^c \sqrt{m_j}}{m} & & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\sum_j^c \sqrt{m_j}}{m} \end{bmatrix}.$$

So

$$\begin{aligned} \mathrm{trace}(S_B) &= (E\alpha)^T \left(Y_0 M^T\right)^T \left(Y_0 M^T\right)(E\alpha) \\ &= (E\alpha)^T \left(Y_0 M^T\right)^T \left(Y_0 M^T\right)(E\alpha) \\ &= \alpha^T \left(Y_0 M^T E\right)^T \left(Y_0 M^T E\right)\alpha \end{aligned} \tag{9.37}$$

Let $X_B = Y_0 M^T E$, then

$$\mathrm{trace}(S_B) = \alpha^T X_B^T X_B \alpha \tag{9.38}$$

Let $\tilde{S}_B = X_B X_B^T$ and $\tilde{S}_T = X_T X_T^T$, then

$$\widetilde{\mathrm{Dis}}(\alpha) = \mathrm{trace}\left(\alpha^T \left(2\tilde{S}_B - \tilde{S}_T\right)\alpha\right) \tag{9.39}$$

Optimizing the kernel with maximum margin criterion aims to maximize the margin of samples in the empirical feature space, that is, to find the optimal α to maximize $\widetilde{\mathrm{Dis}}(\alpha)$. As same as the Fisher criterion method, let $\alpha^T\alpha = 1$, the optimization is defined as

$$\begin{aligned} &\max \quad \alpha^T \left(2\tilde{S}_B - \tilde{S}_T\right)\alpha \\ &\text{subject to} \quad \alpha^T\alpha - 1 = 0 \end{aligned} \tag{9.40}$$

It is easy to know that the optimal expansion coefficient vector α^* is equal to the eigenvector of $2\tilde{S}_B - \tilde{S}_T$ corresponding to the maximal eigenvalue. We use the similar method in [1] to solve the eigenvector and eigenvalue of matrix $2\tilde{S}_B - \tilde{S}_T$ as follows:

$$P^T \tilde{S}_B P = \wedge \tag{9.41}$$

$$P^T \tilde{S}_T P = I \tag{9.42}$$

where $P = \phi\theta^{-1/2}\psi$, θ, and ϕ are the eigenvalue and eigenvector of $\tilde{S}_T$, respectively. ψ is the eigenvalue matrix of $\theta^{-1/2}\phi^T\tilde{S}_B\phi\theta^{-1/2}$. So the column vector of P is the eigenvalue matrix of $2\tilde{S}_B - \tilde{S}_T$ corresponding to the eigenvalue $2 \wedge -I$.

We apply SVD method to calculate the eigenvalue matrix .

The differences between two kernel optimization methods are shown as follows. Solution method: Fisher criterion method use iteration method to solve the solution, while maximum margin criterion method is used to find the optimal solution with the eigenvalue equation. Computation method: FC method costs much more time than MMC method. Moreover, FC method needs to choose the relative parameters in advance, while MMC method needs not to choose the parameters in advance.

9.3 Simulations and Discussion

9.3.1 Experimental Setting and Databases

In this section, we evaluate kernel optimization methods on simulated data, UCI database, ORL database, and Yale database. The general scheme of experiments is shown in Figure 9.2. We evaluate the popular kernel learning methods including sparse KPCA [21], KDA, and KCLPP [10] under kernel optimization. Firstly, on simulated data, we evaluate the data distribution in discriminant with kernel-optimized learning on Fisher and maximum margin criterions. From this set of experiments, the results are shown directly. Secondly, we implement the kernel-optimized learning method on UCI database compared with sparse KPCA method. Finally, we compare KDA and KCLPP with its kernel optimization methods on ORL and Yale databases. The experiments are implemented on a Pentium 3.0 GHz computer with 512-MB RAM and programmed in MATLAB, and the procedural parameters are chosen with cross-validation method.

1. Simulated Gaussian dataset

On the simulated Gaussian dataset, we use two classes of samples for the performance evaluation of kernel optimization. Figure 9.3 describes the 2D data distribution of two classes, and the two classes include 150 and 170 samples under the different parameter of Gaussain function.

2. UCI dataset

On 6 UCI datasets, we evaluate the kernel optimization under KPCA. In the experiments, we randomly choose 20 % of the total number as the training

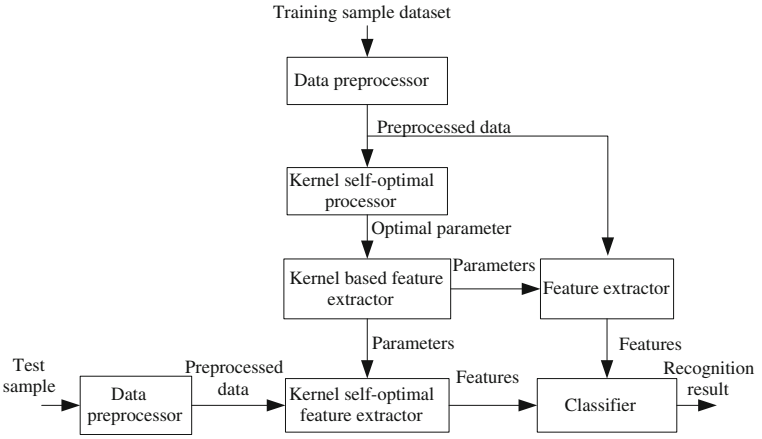
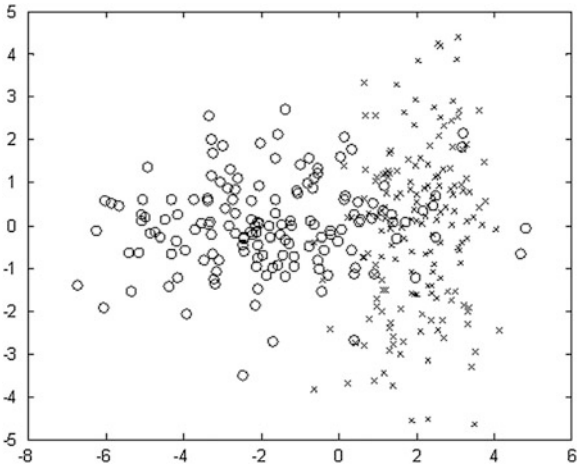


Fig. 9.2 Framework of kernel self-optimization learning

Fig. 9.3 Two classes of two-dimensional data samples with Gaussian distribution



samples in image and splice datasets, and the rest other samples are regarded as the test samples. Moreover, on other 4 datasets, 100 samples are chosen randomly as the training samples, and the rest samples of each dataset are considered as the test training samples.

3. Yale database

The Yale face database [22] was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals in this database. These images are taken under different lighting conditions (left-light, center-light, right-light) and different facial expressions (normal, happy, sad, sleepy, surprised, and wink) and with/without glasses. Some example face images were cropped to the size of 100×100 pixels.

4. ORL database

The ORL face database [6], developed at the Olivetti Research Laboratory, Cambridge, UK, is composed of 400 grayscale images, and each of 40 individuals has 10 images. The variations in the images are across pose, time, and facial expression. The image is resized to 48×48 pixels in this experiment.

5. MIAS database

On MIAS database, we evaluate the medical classification performance on MATLAB platform on the simulation condition. The set of experiments are implemented in a digital mammography database, Mini-MIAS database [23], developed by the Mammography Image Analysis Society. The 1024×1024 pixels of image were achieved through digitizing X-ray films with a Joyce-Loebl scanning microdensitometer to a resolution of 50×50 μm. The experiment framework includes the two key steps of preprocessing and classification. The preprocessing step removes the artifacts and pectoral muscle, and the classification step classifies the test image into three types of fatty, glandular, and dense. The experimental sample database includes 12 fatty images, 14 glandular tissue images, and 16 dense tissue images. The recognition accuracy is evaluated with cross-validation method.

9.3.2 Performance Evaluation on Two Criterions and Four Definitions of $e(x, z_n)$

In this section, we evaluate two criterions, Fisher criterion and maximum margin criterion, and then, we test the feasibility of four methods of defining $e(x, z_n)$, M1, M2, M3, and M4. The experiments are implemented on simulated dataset and two face databases. Gaussian kernel $k(x, y) = e^{-0.001\|x-y\|^2}$ and polynomial kernel $k(x, y) = (x^T y)^2$ are considered as the basic kernel.

On two criterions of solving the expansion coefficients, as shown in Fig. 9.4, the discriminative ability of samples decreases owing to the bad chosen basic kernel, so the kernel optimization is necessary to enhance the discriminative ability of data in the empirical feature space. Figure 9.5 shows the result of Fisher criterion under the different iterations. Figure 9.5a, c, and e are the results under the Gaussian kernels as the basic kernel during kernel optimization, and the rest choose polynomial kernel as the basic kernel. Under the same basic kernel, the different iterations have the different data class discrimination. For examples, as shown in Fig. 9.5b and f, 25 iterations of kernel optimization have less class discrimination compared with 150 iterations. Under the enough number of iterations, Fisher criterion achieves the similar discriminative ability to maximum margin criterion, which is shown in Fig. 9.6. So, iteration number has heavy influences on performance of kernel optimization based on Fisher criterion.

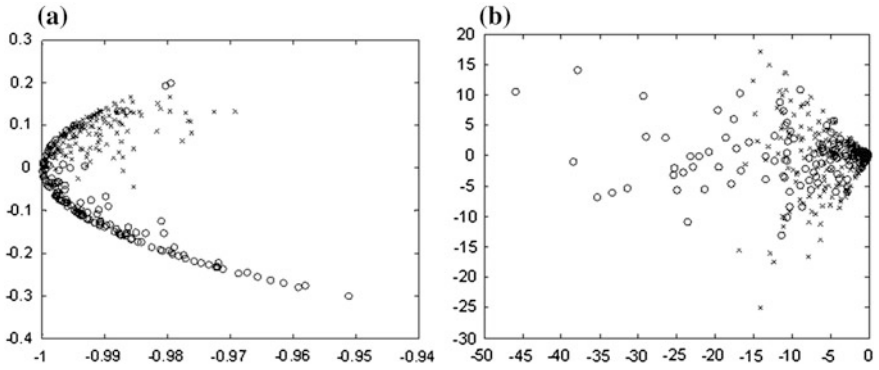


Fig. 9.4 Distribution of data samples in the empirical feature space. **a** Gaussian kernel. **b** Polynomial kernel

We also implement some experiments with the practical face recognition system as shown in Fig. 9.7. The computation efficiency and recognition accuracy are used to evaluate the performance of kernel optimization. The computation efficiency is measured with the time consumption of solving the expansion coefficient vector.

We also evaluate the four definitions of $e(x, z_n)$ in (9.2) marked by M1, M2, M3, and M4, under the framework of Fig. 9.7. Evaluation results are shown in Table 9.1. MMC achieves a higher recognition accuracy and computation efficiency compared with FC under the same method of defining $e(x, z_n)$, M1, M2, M3, and M4. The four methods achieve the similar recognition accuracy but different computation efficiency. M3 and M4 outperform M1 and M2 on the computation efficiency. MMC-based kernel self-optimization method is applied to solve the expansion coefficients [13].

9.3.3 Comprehensive Evaluations on UCI Dataset

We evaluate the kernel optimization on sparse kernel principal component analysis (SKPCA) [21], and SKPCA is formulated with least-squares support vector machine. The main idea of SKPCA is to choose the representative training samples to save the number of training samples. This method only needs little number of training samples for computing the kernel matrix during kernel learning.

For input vector x, the SKPCA-based vector y should be computed as follows:

$$y = B^T V_{zx} \tag{9.43}$$

where $V_{zx} = [g(z_1, x), g(z_2, x), \ldots, g(z_{Nz}, x)]^T$, $g(z_i, x) = k(z_i, x) - \frac{1}{q}\sum_{q=1}^{N} k(z_i, x_q)$, Nz is the number of representative training samples, and B is the projection matrix. Kernel-optimized SKPCA chooses adaptively a few of samples from the training

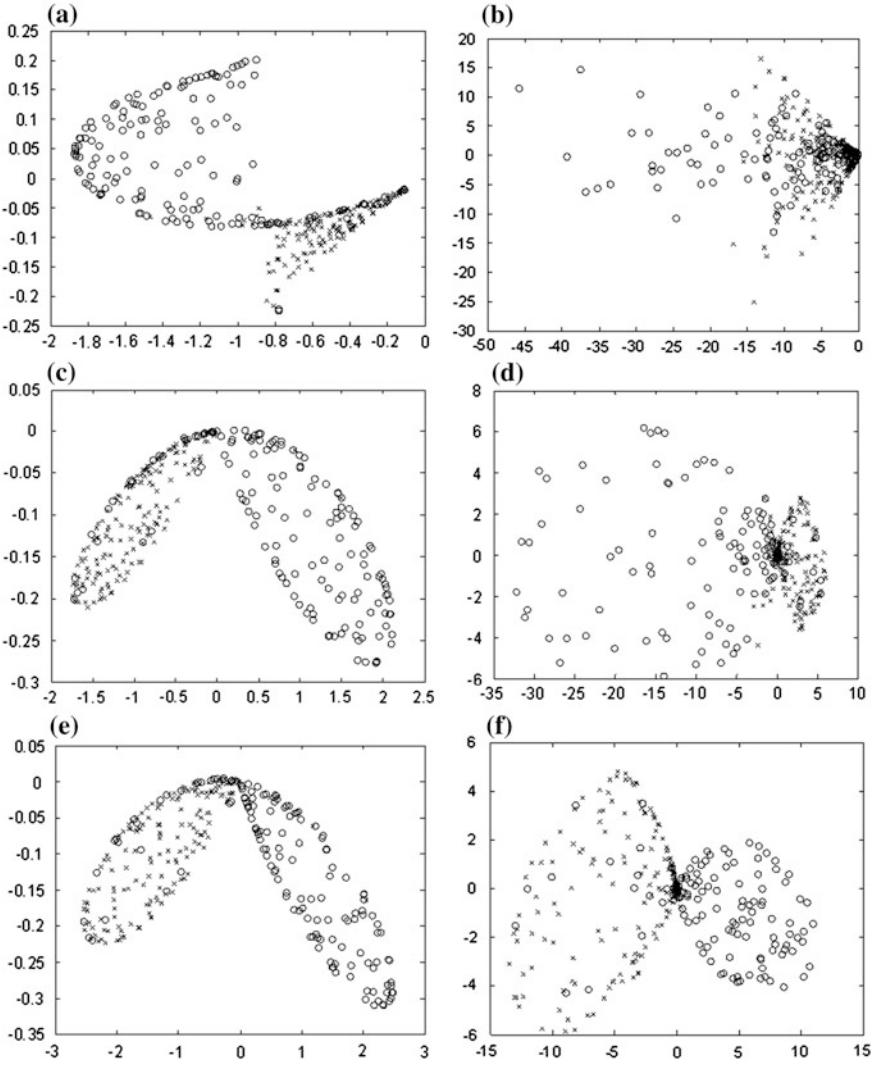


Fig. 9.5 Fisher-criterion-based kernel optimization. **a** Gaussian kernel (25 iterations). **b** Polynomial kernel (25 iterations). **c** Gaussian kernel (75 iterations). **d** Polynomial kernel (75 iterations). **e** Gaussian kernel (150 iterations). **f** Polynomial kernel (150 iterations)

sample set and maintains the maximum recognition performance. The kernel-optimized SKPCA saves much space of storing training samples on computing the kernel matrix with the lower time consumption. In the practical applications, kernel-optimized SKPCA solves the KPCA's limitations on high store space and time consumption. So from the theory viewpoint, this application of

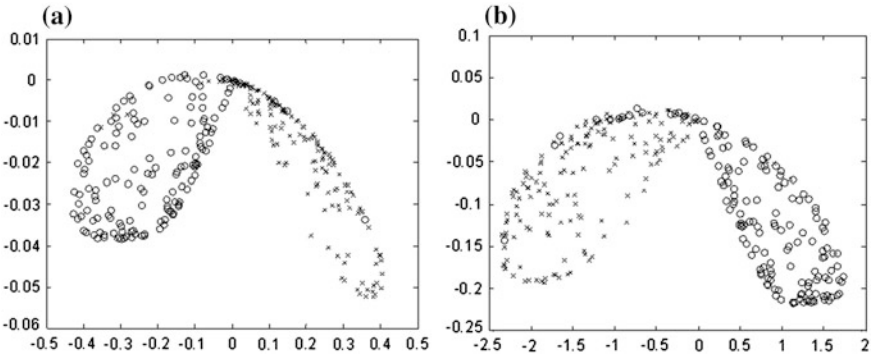


Fig. 9.6 Performance comparison of two algorithms. **a** Maximum margin criterion. **b** Fisher criterion

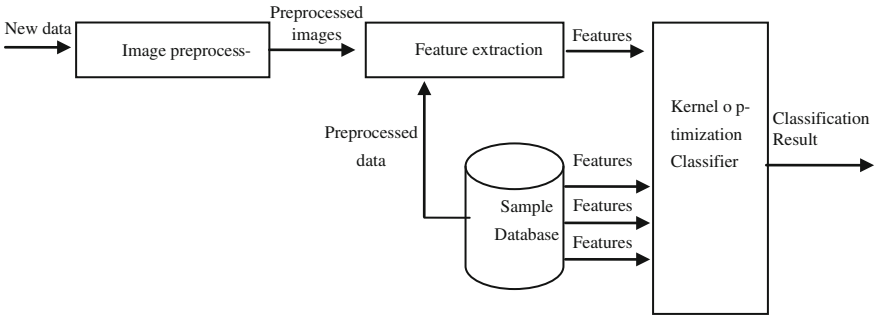


Fig. 9.7 A framework of face recognition system

Table 9.1 Performance of two kernel optimization methods under four extended data-dependent kernel

		ORL				YALE			
Performance	Optimization methods	M1	M2	M3	M4	M1	M2	M3	M4
Recognition accuracy (%)	MMC	93.65	93.40	93.35	93.55	91.87	92.27	92.00	92.27
	FC	92.45	92.15	92.10	92.40	90.00	91.47	90.79	91.87
Computation efficiency (seconds)	MMC	0.05	0.05	0.02	0.02	0.03	0.05	0.02	0.02
	FC	1.50	1.48	0.20	0.19	0.32	0.32	0.08	0.06

kernel-optimized learning is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

In the experiments, we use the sparse analysis to determine some key training samples as the final sample for kernel learning. In our previous work [21], we have concluded the good recognition performance be achieved only using the less size

of training samples. Kernel optimization performs well on recognition performance as shown in Table 9.2. Kernel-optimization-based SKPCA outperforms SKPCA under the same number of training samples. It is feasible to improve the SKPCA with kernel optimization.

Moreover, we also evaluate kernel-optimization-based SKPCA on the practical ORL and YALE databases. The experimental results are shown in Table 9.3, and kernel-optimization-based SKPCA performs better than SKPCA under the same number of training samples. On the computation efficiency, computing the kernel matrix has its heavy time consumption, so only a part of training samples are selected as the representative training samples, which increase computation efficiency. Kernel-optimization-based SKPCA algorithm achieves the higher recognition accuracy than SKPCA owing to its kernel optimization combined with SKPCA, which is adaptive to the applications with the demand of the strict computation efficiency but not strict on recognition.

9.3.4 Comprehensive Evaluations on Yale and ORL Databases

On real Yale database, we evaluate the recognition performance of kernel optimization on KCLPP [10]. The kernel LPP is shown as follows:

$$\min_{w} \; w^T \, KLKw,$$
$$\text{subject to} \quad w^T \, KDKw = 1, \tag{9.44}$$

where L and D are the matrix computed with the input samples and w is the projection vector. Based on KCLPP, we extend it to kernel-optimization-based KCLPP as

$$\min_{w} \; w^T \, K^{(A)} L K^{(A)} w,$$
$$\text{subject to} \quad w^T \, K^{(A)} D K^{(A)} w = 1, \tag{9.45}$$

where w is the optimal projection matrix and $K^{(A)}$ is data-dependent kernel matrix.

Table 9.2 Error rate of kernel optimization on UCI database (%)

Datasets	Training samples	Key training samples	SKPCA [25]	Kernel optimization SKPCA
Banana	400	120	14.2 ± 0.1	13.9 ± 0.2
Image	1,300	180	5.4 ± 0.3	5.1 ± 0.2
F.Solar	666	50	34.2 ± 2.3	32.8 ± 2.1
Splice	1,000	280	9.4 ± 0.9	9.0 ± 0.7
Thyroid	140	30	2.2 ± 0.7	1.2 ± 0.6
Titanic	150	30	24.4 ± 0.4	23.2 ± 0.5

Table 9.3 Performance evaluation of kernel-optimization-based SKPCA on YALE and ORL databases

	YALE database		ORL database	
Methods	Error rate (%)	Training samples	Error rate (%)	Training samples
KPCA	17.8 ± 0.7	75	15.3 ± 0.8	200
SKPCA	20.4 ± 0.8	45 (60 %)	18.4 ± 0.9	120 (60 %)
Kernel-optimized SKPCA	18.7 ± 0.6	45 (60 %)	17.5 ± 0.7	120 (60 %)

In the experiments, we implement the algorithms on five subdatasets SD1, SD2, SD3, SD4, and SD5 for analyzing the influence of pose, illumination, and expression (PIE) (Table 9.4).

The computation efficiency and recognition accuracy are used to evaluate the performance of the proposed algorithm. The optimal procedural parameters of each algorithm were chosen through the cross-validation method. Moreover, we also implement the traditional LPP [10] together with PCA [22], KDA [24], CLPP [10]. Moreover, we also implement fuzzy-based kernel classifier, fuzzy kernel Fisher discriminant (FKFD), through applying fuzzy trick [20, 19] into kernel Fisher classifier. The experimental results are shown in Table 9.5. Kernel-optimized fuzzy-based kernel Fisher classifier achieves the higher recognition accuracy than fuzzy kernel Fisher classifier. On the KCLPP and KDA methods, kernel optimization versions improve the recognition performance.

On ORL database, we implement kernel-optimized learning version of KDA, FKFD, and KCLPP [10, 21] compared with PCA [22], KDA [24], CLPP [10], and KCLPP [10]. The averaged recognition rate is considered as the recognition accuracy. As shown in Table 9.6, the averaged recognition rate of LPP, CLPP, and KCLPP and the proposed kernel-optimized KCLPP are 93.80, 94.80, 96.50, and 98.00 %, respectively. It is feasible to apply kernel optimization method to improve the recognition performance of kernel-based learning.

The feasibility and performance are evaluated on Gaussian data dataset, UCI dataset, and ORL and Yale databases. Firstly, on the simulated Gaussian dataset, the data distribution has the large class discriminant for classification. Fisher and maximum margin criterions have the similar performance for classification under the same experimental condition. Secondly, on the UCI, ORL, and Yale databases, kernel-optimized learning methods can obtain the higher recognition accuracy than the traditional kernel learning methods. Data-dependent kernel-based optimization improves kernel-based learning. Besides the excellent recognition performance,

Table 9.4 Deterministic training and test set on YALE dataset

	SD1	SD2	SD3	SD4	SD5
Training set	1,2,3,4,5	11,1,2,3,4	9,10,11,1,2,	8,9,10,11,1	7,8,9,10,11
Testing set	6,7,8,9,10,11	5,6,7,8,9,10	3,4,5,6,7,8	2,3,4,5,6,7	1,2,3,4,5,6

Note 1 denotes the first image of each person, *2* denotes the second image of each person, and other images are marked with the same ways

Table 9.5 Recognition accuracy on Yale subdatabases (%)

Datasets	PCA [1]	KDA [5]	Kernel-optimized KDA	FKFD	Kernel-optimized FKFD	LPP [13]	CLPP [13]	KCLPP [13]	Kernel-optimized KCLPP
SD1	84.33	94.44	95.22	95.67	96.33	86.33	90.00	94.44	95.67
SD2	86.77	92.22	93.33	93.33	94.33	90.67	91.11	92.22	93.33
SD3	85.33	93.33	94.44	94.22	95.67	88.56	86.67	93.33	94.44
SD4	85.67	93.33	94.67	94.33	95.71	88.89	90.00	93.33	92.33
SD5	88.67	96.67	98.22	97.67	98.87	95.56	93.33	96.67	97.44
Averaged	86.15	94.00	95.18	95.00	96.18	90.00	90.22	93.99	94.62

Table 9.6 Recognition accuracy on ORL subdatabases (%)

Datasets	PCA [1]	KDA [5]	Kernel-optimized KDA	FKFD	Kernel-optimized FKFD	LPP [13]	CLPP [13]	KCLPP[13]	Kernel-optimized KCLPP
SD1	92.00	93.50	94.50	94.00	94.50	95.00	96.00	96.50	98.50
SD2	92.00	93.50	94.50	94.00	94.50	93.50	94.00	95.50	97.50
SD3	93.00	94.00	95.50	94.50	95.00	95.50	97.00	98.50	99.50
SD4	92.00	93.50	94.50	94.00	94.50	93.50	94.50	96.00	97.50
SD5	90.50	91.00	92.50	92.00	92.50	91.50	92.50	96.00	97.00
Averaged	91.90	93.10	94.30	94.20	94.70	93.80	94.80	96.50	98.00

the efficiency of kernel-optimized learning algorithms is still one problem worth to discuss. We also evaluate the efficiency of kernel optimization method. The computation cost is measured with the time of calculating the projection matrices. The experimental result is shown in Table 9.7. The experimental results show that the proposed kernel optimization method is adaptive to the practical applications of high recognition accuracy but low time consumption. The experiments are implemented on face databases, and the largest dimension of kernel matrix is 400 plus 400. The main time consumption comes from kernel matrix computing, so the dimension of matrix has a more influence on time consumption. If the scale of dataset is more than 50,000 points, 20 classes, and the dimension of feature is 200, then the dimension of kernel matrix is 1,000,000 plus 1,000,000. Thus, the computation consuming increases at a very large scale. So the proposed learning method endures time computation problem for a very large scale of dataset.

Table 9.7 Computation cost (seconds) in calculating the projection matrices on Yale database

	KDA	Kernel-optimized KDA	KCLPP	Kernel-optimized KCLPP
ORL	5.7252	6.8345	5.7897	8.7567
YALE	2.3794	3.8305	2.4539	4.8693

9.4 Discussion

In this book, we present a framework of kernel optimization for kernel-based learning. This framework solves the kernel function selection problem endured widely by many kernel learning methods. In the kernel-based system, the data distribution in the nonlinear feature space is determined by kernel mapping. In this framework, two-criterion-based kernel optimization objective functions are applied to achieve the optimized parameters for the good data discriminative distributions in the kernel mapping space. Time consumption is a crucial issue of kernel optimization in the large scale of datasets. The main time consumption lies in kernel matrix computing, so the dimension of matrix has more influence on time consumption. How to improve the computing efficiency is the future work.

References

1. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Netw 12(6):783–789
2. Sharma A, Paliwal KK, Imoto S, Miyano S (2012) Principal component analysis using QR decomposition. Int J Mach Learn Cybern. doi:10.1007/s13042-012-0131-7
3. Chen C, Zhang J, He X, Zhou ZH (2012) Non-parametric kernel learning with robust pairwise constraints. Int J Mach Learn Cybern 3(2):83–96
4. Zhua Q (2010) Reformative nonlinear feature extraction using kernel MSE. Neurocomputing 73(16–18):3334–3337
5. Baudat G, Anouar F (2000) Generalized discriminant analysis using a kernel approach. Neural Comput 12(10):2385–2404
6. Liang Z, Shi P (2005) Uncorrelated discriminant vectors using a kernel method. Pattern Recogn 38:307–310
7. Wang L, Chan KL, Xue P (2005) A criterion for optimizing kernel parameters in KBDA for image retrieval. IEEE Trans Syst, Man and Cybern B: Cybern 35(3):556–562
8. Chen WS, Yuen PC, Huang J, Dai DQ (2005) Kernel machine-based one-parameter regularized Fisher discriminant method for face recognition. IEEE Trans Syst, Man Cybern B: Cybern 35(4):658–669
9. Lu J, Plataniotis KN, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans Neural Netw 14(1):117–226
10. Li JB, Pan JS, Chu SC (2008) Kernel class-wise locality preserving projection. Inf Sci 178(7):1825–1835
11. Feng G, Hu D, Zhang D, Zhou Z (2006) An alternative formulation of kernel LPP with application to image recognition. Neurocomputing 69(13–15):1733–1738
12. Cheng J, Liu Q, Lua H, Chen YW (2005) Supervised kernel locality preserving projections for face recognition. Neurocomputing 67:443–449
13. Huang J, Yuen PC, Chen WS, LaiJH (2004) Kernel subspace LDA with optimized kernel parameters on face recognition. In: Proceedings of the sixth IEEE international conference on automatic face and gesture recognition
14. Zhao H, Sun S, Jing Z, Yang J (2006) Local structure based supervised feature extraction. Pattern Recogn 39(8):1546–1550
15. Pan JS, Li JB, Lu ZM (2008) Adaptive quasiconformal kernel discriminant analysis. Neurocomputing 71(13–15):2754–2760

16. Li JB, Pan JS, Chen SM (2011) Kernel self-optimized locality preserving discriminant analysis for feature extraction and recognition. Neurocomputing 74(17):3019–3027
17. Xiong H, Swamy MN, Ahmad MO (2005) Optimizing the kernel in the empirical feature space. IEEE Trans Neural Netw 16(2):460–474
18. Li JB, Pan JS, Lu ZM (2009) Kernel optimization-based discriminant analysis for face recognition. Neural Comput Appl 18(6):603–612
19. Wang X, Dong C (2009) Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy. IEEE Trans Fuzzy Syst 17(3):556–567
20. Wang X, Hong JR (1999) Learning optimization in simplifying fuzzy rules. Fuzzy Sets Syst 106(3):349–356
21. Li JB, Yu LJ, Sun SH (2011) Refined kernel principal component analysis based feature extraction. Chin J Electron 20(3):467–470
22. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Machine Mach Intell 19(7):711–720
23. Suckling J, Parker J, Dance D, Astley S, Astley I, Hutt I, Boggis C (1994) The mammographic images analysis society digital mammogram database. Exerpta Med 1069:375–378
24. Yang MH (2002) Kernel eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods. In: Proceedings of fifth IEEE international conference on automatic face and gesture recognition, pp 215–220
25. Wang XH, Good WF, Chapman BE, Chang YH, Poller WR, Chang TS, Hardesty LA (2003) Automated assessment of the composition of breast tissue revealed on tissue-thickness-corrected mammography. Am J Roentgenol 180:227–262

Chapter 10
Kernel Construction for Face Recognition

10.1 Introduction

Face recognition and its relative research [1–3] have become the very active research topics in recent years due to its wide applications. An excellent face recognition algorithm should sufficiently consider the following two issues: what features are used to represent a face image and how to classify a new face image based on this representation. So the facial feature extraction plays an important role in face recognition. Among various facial feature extraction methods, the dimensionality reduction technique is exciting since the low-dimensional feature representation with high discriminatory power is very important for facial feature extraction, such as principal component analysis (PCA) and linear discriminant analysis (LDA) [4–6]. Although successful in many cases, these linear methods cannot provide reliable and robust solutions to those face recognition problems with complex face variations since the distribution of face images under a perceivable variation in viewpoint, illumination, or facial expression is highly nonlinear and complex. Recently, researchers applied kernel machine techniques to solve the nonlinear problem successfully [7–9], and accordingly some kernel-based methods are developed for face recognition [10–14]. But current kernel-based facial feature extraction methods face the following problems. (1) Current face recognition methods are based on image or video, while the current popular kernels need format of the input data as a vector. Thus, kernel-based facial feature extraction causes the large storage requirements and the large computational effort for transforming images to vectors owing to its viewing images as vectors. (2) Different kernels can cause the different RKHS in which the data have different class discrimination, so the selection of kernels will influence the recognition performance of the kernel-based methods. And the inappropriate selection of kernels will decrease the performance. But unfortunately the geometrical structures of the data in the feature space will not be changeable when we only change the parameter of the kernel.

In this chapter, a novel kernel named Adaptive Data-dependent Matrix Norm-Based Gaussian Kernel (ADM-Gaussian kernel) is proposed in this chapter.

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2_10,

Firstly, we create a novel matrix norm-based Gaussian kernel, which views images as matrices for facial feature extraction, as the basic kernel of data-dependent kernel, while the data-dependent kernel can change the geometrical structures of the data with different expansion coefficients. And then we apply the maximum margin criterion to solve the adaptive expansion coefficients of ADM-Gaussian kernel which leads the largest class discrimination in the feature space.

In this chapter, we propose a novel kernel named Adaptive Data-dependent Matrix Norm-Based Gaussian Kernel (ADM-Gaussian kernel) for facial feature extraction. As a popular facial feature extraction method for face recognition, the current kernel method endures some problems. Firstly, the face image must be transformed to the vector, which leads to the large storage requirements and the large computational effort, and secondly, since the different geometrical structures lead to the different class discrimination of the data in the feature space, the performance of the kernel method is influenced when kernels is inappropriately selected. In order to solve these problems, firstly, we create a novel matrix norm-based Gaussian kernel which views images as matrices for facial feature extraction, which is the basic kernel for the data-dependent kernel. Secondly, we apply a novel maximum margin criterion to seek the adaptive expansion coefficients of the data-dependent kernel, which leads to the largest class discrimination of the data in the feature space. Experiments on ORL and Yale databases demonstrate the effectiveness of the proposed algorithm.

10.2 Matrix Norm-Based Gaussian Kernel

In this section, firstly, we introduce the data-dependent kernel-based on vectors, and then we extend it to the version of matrices. Secondly, we introduce the theoretical analysis of matrix norm-based Gaussian kernel, and finally, we apply the maximum margin criterion to seek the adaptive expansion coefficients of the data-dependent matrix norm-based Gaussian kernel.

10.2.1 Data-Dependent Kernel

Data-dependent kernel with a general geometrical structure is applied to create a new kernel in this chapter. Given a basic kernel $k_b(x, y)$, its data-dependent kernel $k_d(x, y)$ can be defined as follows.

$$k_d(x, y) = f(x)f(y)k_b(x, y) \tag{10.1}$$

where $f(x)$ is a positive real-valued function x, which is defined as follows.

$$f(x) = b_0 + \sum_{n=1}^{N} b_n e(x, \tilde{x}_n) \tag{10.2}$$

In the previous work in [15], Amari and Wu expanded the spatial resolution in the margin of a SVM by using $f(x) = \sum i \in SVa_i e^{-\delta\|x-\tilde{x}_i\|^2}$, where $\tilde{x}_i$ is the ith support vector, SV is a set of support vector, a_i is a positive number representing the contribution of $\tilde{x}_i$, and δ is a free parameter.

We extend it to the matrix version and propose the Adaptive Data-dependent Matrix Norm-Based Gaussian Kernel (ADM-Gaussian kernel) as follows. Supposed that $k_b(X, Y)$ is so-called matrix norm-based Gaussian kernel (M-Gaussian kernel) as the basic kernel, and $k_d(X, Y)$ is a data-dependent matrix norm-based Gaussian kernel. Then data-dependent matrix norm-based Gaussian kernel is defined as follows.

$$k_d(X, Y) = f(X)f(Y)k_b(X, Y) \tag{10.3}$$

where $f(X)$ is a positive real-valued function X,

$$f(X) = b_0 + \sum_{n=1}^{N_{XV}} b_n e\left(X, \widetilde{X}_n\right) \tag{10.4}$$

where $e\left(X, \widetilde{X}_n\right)(1 \le n \le N_{XM})$ is defined as follows.

$$e\left(X, \widetilde{X}_n\right) = \exp\left(-\delta \sum_{j=1}^{N} \left(\sum_{i=1}^{M} \left(x_{ij} - \widetilde{x_{ij}}\right)^2\right)^{1/2}\right) \tag{10.5}$$

where $\widetilde{x_{ij}}(i = 1, 2, \ldots, M, j = 1, 2, \ldots, N)$ are the elements of matrix $\widetilde{x_n}(n = 1, 2, \ldots, N_{XM})$, and δ is a free parameter, and $\widetilde{X}_n,\ 1 \le n \le N_{XM}$ are called the "expansion matrices (XMs)" in this chapter, N_{XM} is the number of XMs, and $b_i \in R$ is the "expansion coefficient" associated with $\widetilde{X}_n$. The $\widetilde{X}_n,\ 1 \le n \le N_{XM}$, for its vector version, have different notations in the different kernel learning algorithms.

10.2.2 Matrix Norm-Based Gaussian Kernel

Given n samples $X_p^q \left(X_p^q \in \mathbb{R}^{M \times N}\right),\ (p = 1, 2, \ldots, L,\ q = 1, 2, \ldots, n_p)$ where n_p denotes the number of samples in the pth class and L denote the number of the classes. M-Gaussian kernel $k_b(X, Y)$ is defined as follows.

$$k(X,Y)=\exp\left(-\frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}\left(x_{ij}-y_{ij}\right)^2\right)^{1/2}}{2\sigma^2}\right)\ (\sigma>0) \tag{10.6}$$

where $X=\left[x_{ij}\right]_{i=1,2,\ldots,M;\ j=1,2,\ldots,N}$ and $Y=\left[y_{ij}\right]_{i=1,2,\ldots,M;\ j=1,2,\ldots,N}$ denote two sample matrices. Now we want to prove that $k(X,Y)=e^{-\frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}\left(x_{ij}-y_{ij}\right)^2\right)^{1/2}}{2\sigma^2}}$ is a kernel function. Kernel function can be defined in various ways. In most cases, however, kernel means a function whose value only depends on a distance between the input data, which may be vectors.

It is a sufficient and necessary condition for a symmetric function to be a kernel function that its Gram matrix is positive semi-definite [16]. Given a finite data set $X=\{x_1,x_2,\ldots,x_N\}$ in the input space and a function $k(\cdot,\cdot)$, the $N\times N$ matrix K with elements $K_{ij}=k\left(x_i,x_j\right)$ is called Gram matrix of $k(\cdot,\cdot)$ with respect to $x_1,x_2,\ldots,x_N$. And it is easy to know that $k(X,Y)=e^{-\frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}\left(x_{ij}-y_{ij}\right)^2\right)^{1/2}}{2\sigma^2}}$ is a symmetric function. The matrix K, which is derived from the $k(X,Y)$, is positive and definite. While it is easy to know that $F(X)=\sum_{j=1}^{N}\left(\sum_{i=1}^{M}x_{ij}^2\right)^{\frac{1}{2}}$, $\left(X=\left[x_{ij}\right]_{i=1,2,\ldots,M;\ j=1,2,\ldots,N}\right)$ is a matrix norm. $k(X,Y)=e^{-\frac{\sum_{j=1}^{N}\left(\sum_{i=1}^{M}\left(x_{ij}-y_{ij}\right)^2\right)^{1/2}}{2\sigma^2}}$ is derived form the matrix norm, so we can call it a matrix norm-based Gaussian kernel.

Gaussian kernel denotes the distribution of similarity between two vectors. Similarly M-Gaussian kernel also denotes the distribution of similarity between two matrices. M-Gaussian kernel views an image as a matrix, which enhances the computation efficiency without influencing the performance of kernel-based method.

10.3 Adaptive Matrix-Based Gaussian Kernel

In this section, our goal is to seek the optimal expansion coefficients for the data-dependent matrix norm-based Gaussian kernel (ADM-Gaussian kernel), and the ADM-Gaussian kernel is adaptive to the input data in the feature space. The data in the feature space have the largest class discrimination with the ADM-Gaussian kernel.

10.3.1 Theory Deviation

Firstly, our goal is to select the free parameter δ and the expansion matrices $\tilde{X}_n,\ 1 \le n \le N_{XM}$. In this chapter, we select the mean of the class as the expansion matrix. That is, $N_{XM} = L$. Let $\overline{X}_n$ denotes the mean of the *nth* class, then

$$e\left(X, \widetilde{X}_n\right) = e\left(X, \overline{X}_n\right) = \exp\left(-\delta \sum_{j=1}^{N} \left(\sum_{i-1}^{M} \left(x_{ij} - \overline{x_{ij}}\right)^2\right)^{1/2}\right) \tag{10.7}$$

where $\overline{x_{ij}} (i = 1, 2, \ldots, M, j = 1, 2, \ldots, N)$ are the elements of matrix $\overline{X}_n$ $(n = 1, 2, \ldots, L)$.

After selecting the expansion vectors and the free parameter, our goal is to find the expansion coefficients varied with the input data to optimize the kernel. According to the equation (2), given one free parameter δ and the expansion vectors $\{\tilde{x}_i\}_{i=1,2,\ldots,N_{XV_S}}$, we create a matrix as follows.

$$E = \begin{bmatrix} 1 & e\left(X_1, \tilde{X}_1\right) & \cdots & e\left(X_1, \tilde{X}_{N_{XM_S}}\right) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e\left(X_M, \tilde{X}_1\right) & \cdots & e\left(X_M, \tilde{X}_{N_{XM_S}}\right) \end{bmatrix} \tag{10.8}$$

Let $\beta = [b_0, b_1, b_2, \ldots, b_{N_{XM}}]^T$ and $\Lambda = \mathrm{diag}(f(X_1), f(X_2), \ldots, f(X_n))$, and according to the equation (2), we obtain

$$\Lambda 1_n = E\beta \tag{10.9}$$

where 1_n is a n-dimensional vector whose entries equal to unity.

Proposition 1 *Let K_b and K_d denote the basic M-Gaussian kernel matrix and ADM-Gaussian kernel matrix, respectively, then $K_d = \Lambda k_b \Lambda$.*

Proof Since $K_b = \left[k_b\left(X_i, X_j\right)\right]_{n\times n}$ and $K_d = \left[k_d\left(X_i, X_j\right)\right]_{n\times n}$, according to equation (1), we can obtain

$$k_d\left(X_i, X_j\right) = f(X_i) f\left(X_j\right) k_b\left(X_i, X_j\right) \tag{10.10}$$

And

$$K_d = \left[k_d\left(X_i, X_j\right)\right]_{n\times n} = \left[f(X_i) f\left(X_j\right) k_b\left(X_i, X_j\right)\right]_{n\times n} \tag{10.11}$$

Hence, $K_d = \Lambda k_b \Lambda$ □

Now our goal is to create a constrained optimization function to seek an optimal expansion coefficient vector β. In this chapter, we apply the maximum margin criterion to solve the expansion coefficients. We maximize the class discrimination in the high-dimensional feature space by maximizing the average margin between different classes which is widely used as maximum margin criterion for feature

extraction [17]. The average margin between two classes c_i and c_j in feature space can be defined as follows.

$$\mathrm{Dis} = \frac{1}{2n}\sum_{i=1}^{L}\sum_{j=1}^{L} n_i n_j d\left(c_i, c_j\right) \tag{10.12}$$

where $d\left(c_i, c_j\right) = d\left(m_i^{\Phi}, m_j^{\Phi}\right) - S(c_i) - S\left(c_j\right),\ i,j = 1,2,\ldots,L$ denotes the margin between any two classes, and $S(c_i),\ i = 1,2,\ldots,L$ is the measure of the scatter of the class $c_i,\ i = 1,2,\ldots,L$, and $d\left(m_i^{\Phi}, m_j^{\Phi}\right),\ i,j = 1,2,\ldots,L$ is the distance between the means of two classes. Let $S_i^{\Phi},\ i = 1,2,\ldots,L$ denote the within-class scatter matrix of class i, which is defined as follows.

$$tr\left(S_i^{\Phi}\right) = \frac{1}{n_i}\sum_{p=1}^{n_i}\left(\Phi\left(x_i^p\right) - m_i^{\Phi}\right)^T\left(\Phi\left(x_i^p\right) - m_i^{\Phi}\right) \tag{10.13}$$

and $tr\left(S_i^{\Phi}\right)$ measures the scatter of the class i, that is, $S(c_i) = tr\left(S_i^{\Phi}\right)$, $i = 1,2,\ldots,L$.

Proposition 2 *Let* $tr\left(S_B^{\Phi}\right)$ *and* $tr\left(S_W^{\Phi}\right)$ *denote the trace of between-class scatter matrix and within classes scatter matrix, respectively, then* $\mathrm{Dis} = tr\left(S_B^{\Phi}\right) - tr\left(S_W^{\Phi}\right)$ □

Proposition 3 Assume $Kc_{ij}\ (i,j = 1,2,\ldots,L)$ *is kernel matrix calculated with the i th and j th class samples and kernel matrix* K_{total} *with its elements* K_{ij}. *Let* $M = 2 * \mathrm{diag}\left(\frac{1}{n_1}Kc_{11}, \frac{1}{n_2}Kc_{22}\ldots, \frac{1}{n_L}Kc_{LL}\right) - \mathrm{diag}(K_{11}, K_{22}, \ldots, K_{nn}) - \frac{1}{n}K_{total}$, *then* $tr\left(S_B^{\Phi}\right) - tr\left(S_W^{\Phi}\right) = 1_n^T M 1_n$.

Detailed proof of the Proposition 2 and 3 can be found in the our previous work [18]. According to Proposition 2 and 3, we can obtain

$$\mathrm{Dis} = 1_n^T M 1_n \tag{10.14}$$

Simultaneously, according to Proposition 1,we can acquire

$$\widetilde{M} = \Lambda M \Lambda \tag{10.15}$$

where $\widetilde{M} = 2 * \mathrm{diag}\left(\frac{1}{n_1}\tilde{K}_{11}, \frac{1}{n_2}\tilde{K}_{22}\ldots, \frac{1}{n_L}\tilde{K}_{LL}\right) - \mathrm{diag}\left(\tilde{K}_{11}, \tilde{K}_{22}, \ldots, \tilde{K}_{nn}\right) - \frac{1}{n}\tilde{K}_{\mathrm{total}}$ and $\tilde{K}_{ij}\ (i,j = 1,2,\ldots,L)$ is calculated by the ith and jth class of samples with the data-dependent kernel, and $\tilde{K}_{\mathrm{total}}$ represents the kernel matrix with its elements $\tilde{k}_{pq}\ (p,q = 1,2,\ldots,n)$ which is calculated by pth and qth samples with adaptive data-dependent kernel. Thus, when a data-dependent kernel is selected as the general kernel, $\widetilde{\mathrm{Dis}}$ is obtained as follows.

$$\widetilde{\mathrm{Dis}} = 1_n^T \Lambda M \Lambda 1_n^T \tag{10.16}$$

The above equation can be written as follows.

$$\widetilde{\mathrm{Dis}} = \beta^T E^T M E \beta \tag{10.17}$$

Given a basic kernel $k(x, y)$ and relative data-dependent kernel coefficients, $E^T ME$ is a constant matrix, so $\widetilde{\mathrm{Dis}}$ is a function with its variable β. So it is reasonable to seek the optimal expansion coefficient vector β by maximizing $\widetilde{\mathrm{Dis}}$. Now we create an optimization function constrained by the unit vector β, i.e., $\beta^T \beta = 1$ as follows.

$$\begin{aligned} &\max\ \beta^T E^T M E \beta \\ &\text{subject to } \beta^T \beta - 1 = 0 \end{aligned} \tag{10.18}$$

The solution of the above constrained optimization problem can often be found by using the so-called Lagrangian method. We define the Lagrangian as

$$L(\beta, \lambda) = \beta^T E^T M E \beta - \lambda\left(\beta^T \beta - 1\right) \tag{10.19}$$

with the parameter λ. The Lagrangian L must be maximized with respect to λ and β, and the derivatives of L with respect to β must vanish, that is,

$$\frac{\partial L(\beta, \lambda)}{\partial \beta} = \left(E^T ME - \lambda I\right)\beta \tag{10.20}$$

And

$$\frac{\partial L(\beta, \lambda)}{\partial \beta} = 0 \tag{10.21}$$

Hence,

$$E^T ME \beta = \lambda \beta \tag{10.22}$$

10.3.2 Algorithm Procedure

So the problem of solving the constrained optimization function is transformed to the problem of solving eigenvalue equation shown in (10.22). We can obtain the optimal expansion coefficient vector β^*, that is, the eigenvector of $E^T ME$ corresponding to the largest eigenvalue. It is easy to see that the data-dependent kernel with β^* is adaptive to the input data, which leads to the best class discrimination in feature space for given input data.

The procedure of creating the ADM-Gaussian kernel can be described as follows.

Step 1. Compute the basic M-Gaussian kernel matrix $K_b = \left[k_b\left(X_i, X_j\right)\right]_{n\times n}$with the formulation (6).

Step 2. Compute the matrix E and M with the formulation (8) and the proposition 2.
Step 3. Obtain the adaptive expansion coefficients vector β^* by solving Eq. (10.22).
Step 4. Calculate the ADM-Gaussian kernel matrix with the $k_d(X, Y)$ with the optimal expansion coefficients vector β^*.

10.4 Experimental Results

10.4.1 Experimental Setting

We implement KPCA with Gaussian kernel, M-Gaussian kernel, and ADM-Gaussian kernel on the two face databases, i.e., ORL face database [19] and Yale face database [1]. We carry out the experiments with two parts as follows: (1) Model selection—selecting the optimal parameters of three Gaussian kernels and the free parameter of the data-dependent kernel for Gaussian kernel, M-Gaussian kernel, and ADM-Gaussian kernel. (2) Performance evaluation—comparing the recognition performance of KPCA with Gaussian kernel, M-Gaussian kernel, and ADM-Gaussian kernel.

In our experiments, we implement our algorithm in the two face databases, ORL face database[19] and Yale face database[1]. The ORL face database, developed at the Olivetti Research Laboratory, Cambridge, U.K., is composed of 400 grayscale images with 10 images for each of 40 individuals. The variations of the images are across pose, time, and facial expression. The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 grayscale images of 15 individuals. These images are taken under different lighting condition (left-light, center-light, and right-light), and different facial expression (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses.

10.4.2 Results

In this section, our goal is to select the kernel parameters of three Gaussian kernels and the free parameter of the data-dependent kernel for Gaussian kernel, M-Gaussian kernel, and ADM-Gaussian kernel. We select the following parameters for selection of Gaussian kernel and the free parameter for M-Gaussian kernel, $\sigma^2 = 1 \times 10^5$, $\sigma^2 = 1 \times 10^6$, $\sigma^2 = 1 \times 10^7$, and $\sigma^2 = 1 \times 10^8$ for M-Gaussian kernel parameter, the free parameter of the data-dependent kernel, $\delta = 1 \times 10^5$, $\delta = 1 \times 10^6$, $\delta = 1 \times 10^7$, $\sigma^2 = 1 \times 10^8$, $\delta = 1 \times 10^9$, and $\delta = 1 \times 10^{10}$. Moreover, the dimension of the feature vector is set to 140 for ORL face database and 70 for Yale face base. From the experimental results, we find

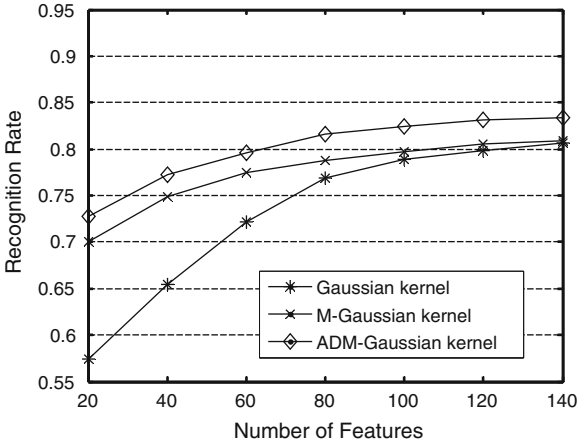


Fig. 10.1 Performance on ORL face database

that the higher recognition rate can be obtained under the following parameters, $\sigma^2 = 1 \times 10^8$ and $\delta = 1 \times 10^5$ for ORL face database and $\sigma^2 = 1 \times 10^8$ and $\delta = 1 \times 10^7$ for Yale face database. After selecting the parameters for AMD-Gaussian kernel, we select the Gaussian parameter for Gaussian kernel and M-Gaussian kernel. $\sigma^2 = 1 \times 10^5$, $\sigma^2 = 1 \times 10^6$, $\sigma^2 = 1 \times 10^7$, and $\sigma^2 = 1 \times 10^8$ are selected to test the performance. From the experiments, we find that, on ORL face database $\sigma^2 = 1 \times 10^6$ is selected for Gaussian, and $\sigma^2 = 1 \times 10^5$ is selected for M-Gaussian kernel. And $\sigma^2 = 1 \times 10^8$ is selected for Gaussian, and $\sigma^2 = 1 \times 10^5$ is selected for M-Gaussian kernel on Yale face database. All these parameters are used in the next section.

In this section, we evaluate the performance the three kinds of Gaussian kernel-based KPCA on the ORL face database and Yale face database. In these experiments, we implement KPCA with the optimal parameters which are selected in the last section. We evaluate the performance of the algorithms with the recognition accuracy with different dimension of features, and as shown in Fig. 10.1 and Fig. 10.2, we can obtain the highest recognition rate of ADM-Gaussian kernel-based KPCA, which is higher than M-Gaussian kernel-based KPCA and Gaussian kernel based on KPCA. Moreover, the higher recognition rate is obtained with M-Gaussian kernel compared with Gaussian kernel.

M-Gaussian kernel achieves the higher recognition accuracy than the traditional Gaussian kernel. Because the ADM-Gaussian kernel is more adaptive to the input data than M-Gaussian kernel, it achieves the higher recognition accuracy than M-Gaussian kernel. All these experimental results are obtained under the optimal parameters, and the selection of optimal parameters of the original Gaussian kernel will influence the performance the kernel. But the ADM-Gaussian will decrease the influence of the parameter selection by its adaptability to the input data.

A novel kernel named Adaptive Data-dependent Matrix Norm-Based Gaussian Kernel (ADM-Gaussian kernel) is proposed for facial feature extraction.

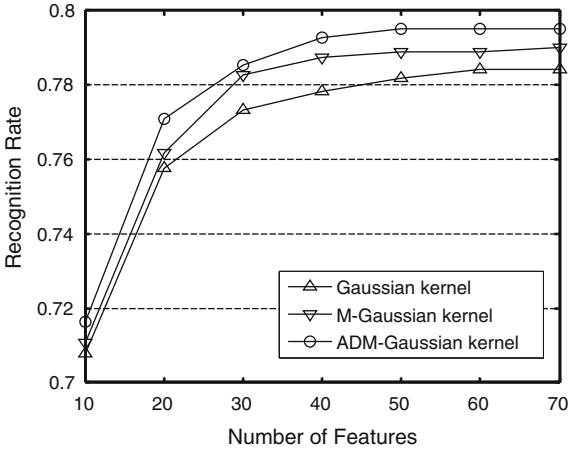


Fig. 10.2 Performance on Yale face database

The ADM-Gaussian kernel views images as matrices, which saves the storage and increase the computational efficiency of feature extraction. Adaptive expansion coefficients of ADM-Gaussian kernel are obtained with the maximum margin criterion, which leads to the largest class discrimination of the data in the feature space. The results, evaluated on two popular databases, suggest that the proposed kernel is superior to the current kernel. In the future, we intend to apply the ADM-Gaussian kernel to other areas, such as content-based image indexing and retrieval as well as video and audio classification.

References

1. Zhe-Ming Lu, Xiu-Na Xu, Pan Jeng-Shyang (2006) Face Detection Based on Vector Quantization in Color Images. International Journal of Innovative Computing, Information and Control. 2(3):667–672
2. Qiao Yu-Long, Zhe-Ming Lu, Pan Jeng-Shyang, Sun Sheng-He (2006) Spline Wavelets Based Texture Features for Image Retrieval. International Journal of Innovative Computing, Information and Control. 2(3):653–658
3. Zhe-Ming Lu, Li Su-Zhi, Burkhardt Hans (2006) "A Content-based Image Retrieval Scheme in JPEG Compressed Domain"., International Journal of Innovative Computing. Inf Control 2(4):831–839
4. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. IEEE Trans. Pattern Analysis and Machine Intelligence 19(7):711–720
5. A.U. Batur and M.H. Hayes, "Linear Subspace for Illumination Robust Face Recognition," Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition, Dec. 2001
6. Martinez AM, Kak AC (2001) PCA versus LDA. IEEE Trans Pattern Analysis and Machine Intelligence 23(2):228–233
7. Schölkopf B, Burges C, Smola AJ (1999) Advances in Kernel Methods—Support Vector Learning. MIT Press, Cambridge, MA

8. Ruiz A, López de Teruel PE (2001) Nonlinear kernel-based statistical pattern analysis. IEEE Trans Neural Networks 12:16–32
9. Müller KR, Mika S, Rätsch G, Tsuda K, Schölkopf B (2001) An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 12:181–201
10. Scholkopf B, Smola A, Muller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
11. Liu Qingshan, Hanqing Lu, Ma Songde (2004) Improving Kernel Fisher Discriminant Analysis for Face Recognition. IEEE Trans Pattern Analysis and Machine Intelligence 14(1):42–49
12. H. Gupta, A. K. Agrawal, T. Pruthi, C. Shekhar, and R. Chellappa, "An experiment evaluation of linear and kernel-based methods for face recognition," presented at the IEEE Workshop on Applications on Computer Vision, Dec. 2002
13. M. H. Yang, "Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods," in Proc. 5th IEEE Int. Conf. Automatic Face and Gesture Recognition, pp. 215–220, May 2002
14. Lu JW, Plataniotis K, Venetsanopoulos AN (2003) Face recognition using kernel direct discriminant analysis algorithms. IEEE Trans. Neural Network 14(1):117–126
15. Amari S, Wu S (1999) Improving support vector machine classifiers by modifying kernel functions. Neural Network 12(6):783–789
16. Cristianini N, Shawe-Taylor J (2000) An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge Univ Press, Cambridge
17. Li Haifeng, Jiang Tao, Zhang Keshu (2006) Efficient and Robust Feature Extraction by Maximum Margin Criterion. IEEE Trans Neural Networks 17(1):157–165
18. Jun-Bao Li, Jeng-Shyang Pan, Zhe-Ming Lu and Bin-Yih Liao, "Data-Dependent Kernel Discriminant Analysis for Feature Extraction and Classification", Proceedings of the (2006) IEEE International Conference on Information AcquisitionAugust 20–23, 2006. Weihai, Shandong, China, pp 1263–1268
19. Ferdinando Samaria, Andy Harter, "Parameterisation of a Stochastic Model for Human Face Identification", Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL, December 1994

Index

J.-B. Li et al., *Kernel Learning Algorithms for Face Recognition*,
DOI: 10.1007/978-1-4614-0161-2,