Neurocomputing 71 (2008) 1850–1856

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A new extension of kernel feature and its application for visual recognition

Qingshan Liu [a,b,*], Hongliang Jin [a,b], Xiaoou Tang [b], Hanqing Lu [a], Songde Ma [a]

[a] National Laboratory of Pattern Recognition, Institute of Automation, CAS, China
[b] Department of Information Engineering, The Chinese University of Hong Kong, China

ARTICLE INFO

Available online 29 February 2008

Keywords:
Kernel feature
Discriminant analysis
Visual recognition

ABSTRACT

In this paper, we first conceive a new perception of the kernel feature. The kernel subspace methods can be regarded as two independent steps: an explicit kernel feature extraction step and a linear subspace analysis step on the extracted kernel features. The kernel feature vector of an image is composed of dot products between the image and all the training images using nonlinear dot product kernel. Then, based on this perception, we further extend the kernel feature vector of an image to a kernel feature matrix for visual recognition. This extension takes different representation cues of images into account, respectively, while only global average information is used in the traditional kernel methods. From the view of dot product as similarity, this extension means using multiple similarities to measure two images, which is more accordant to human vision. In order to efficiently deal with the problem of numerical computation, a matrix-based kernel discriminant analysis algorithm is employed to learn discriminating kernel features for visual recognition. Experiments on the FERET face database, the COIL-100 object database, and the Wang's nature image database show the advantage of the proposed method.

1. Introduction

Principal component analysis (PCA) [12,19] and linear discriminant analysis (LDA) [1,27] are two classical linear subspace methods, and they are widely used for feature extraction. The idea of PCA is to generate a set of orthonormal projections aiming at maximizing the covariance over all the samples. LDA seeks for a linear transformation, which maximizes the between-class scatter and minimizes the within-class scatter. Thus, PCA is optimal for reconstruction, while LDA aims at better discrimination. However, their linear properties limit their performance in many practical applications with complicated nonlinear variations.

Recently, the kernel methods have attracted much attention due to their good nonlinear properties [14,16,25]. The kernel methods first map the input data into an implicit feature space F with a nonlinear mapping, and the data are analyzed in F to get a nonlinear representation. Kernel PCA (KPCA) [16] and kernel discriminant analysis (KDA) [25] are popular nonlinear subspace methods and are widely used in computer vision and pattern recognition [5,6,11,24], in which PCA and LDA are performed to analyze the implicit features in F to extract nonlinear principal components and discriminating features, respectively. In implementation, a nonlinear dot product kernel is introduced to avoid the computation of implicit features, which is defined to calculate the dot product between two implicit feature vectors.

In this paper, we first give a new perception of the kernel feature. The kernel methods are divided into two explicit and independent steps: an explicit kernel feature extraction step and data analysis step (such as linear subspace analysis in the kernel subspace learning [16,25]) on the extracted kernel features. The kernel feature vector of an image consists of dot products between the image and all the training images using nonlinear dot product kernel. Thus, in a sense, the training samples can be regarded as a reference set in the kernel methods. Since this perception defines explicit kernel features, it is easy to manipulate the kernel methods. Here, we further extend the kernel feature vector of an image to a kernel feature matrix for visual recognition. To better capture the complex structure and appearance of visual objects, we often represent the objects with various cues, such as color, texture, local components, and so on. Assuming that we have p different representation cues and N training images, we perform p dot product kernels on these cues, respectively. Then we have an $N \times p$-dimensional matrix kernel feature for an image. Compared with the traditional kernel methods using only global average information, this extension considers different representation cues of images, respectively. From the view of dot product as similarity, the traditional kernel feature only gives an overall similarity with a single dot product, while this extension means

* Corresponding author at: National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China. Tel.: +86 10 62632267; fax: +86 10 62551993.
E-mail address: qsliu@nlpr.ia.ac.cn (Q. Liu).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.10.024

using multiple similarities to measure two images, which is more accordant to human vision. When our eyes discriminate two objects, we are often told the similarities between them from various cues. So we also call the extended kernel feature a multiple similarities-based kernel feature. However, this extended kernel feature brings with it a problem of numerical computation, because most of the popular data analysis methods can only handle the vector-based data. In order to efficiently deal with this problem, a matrix-based KDA (MKDA) algorithm is introduced to learn the discriminating kernel features for visual recognition. This extension is different from couple KDA (CKDA) in [23]. The former is based on an explicit kernel matrix feature, while CKDA is still based on implicit feature space. Moreover, our work aims to consider multiple visual cues, respectively, but CKDA uses multiple kernels for a single image cue. However, because the kernel mapping of each kernel is implicit, simply putting them together and regarding them as a matrix are practically unreasonable. The proposed method gives an explicit interpretation for the kernel matrix feature. In our experiments, we conduct the experiments on the FERET face database [15], the COIL-100 object database [13], and the Wang nature image dataset [2] to test the proposed method. The experimental results show that the proposed method has an encouraging performance.

The rest of the paper is arranged as follows: We first briefly review the kernel-based subspace learning and present a new perception of the kernel feature in Section 2. The new extension of the kernel feature for visual recognition, i.e., multiple similarities-based kernel features for visual recognition, is addressed in Section 3. Experiments are reported in Section 4, and finally conclusions are drawn in Section 5.

2. A new perception of the kernel feature

Before presenting our perception of the kernel feature, we give a short review of KPCA and KDA to show our observation. The ideas of KPCA and KDA are first to map the input data into an implicit feature space F by a nonlinear mapping ϕ, and then PCA and LDA are performed, respectively, in F to get the nonlinear principal components and discriminating features of the input data [16,25]. It is unnecessary to know the implicit feature vector $\phi(x)$ explicitly, and we only need to calculate the dot product between two implicit feature vectors with a Mercer dot product kernel, such as the Gaussian kernel used in the paper, $k(x,y) = (\phi(x)\cdot\phi(y)) = \exp(-\gamma||(x-y)/\sigma||^2)$.

For the following analysis, we define some symbols first. $X = \{x_1, x_2, \ldots, x_N\}$ is the training set with N images and C classes. Each class has N_c samples, and X_c represents the sample set of the cth class. Define the dot product matrix $K = [K_1, K_2, \ldots, K_N]$, where the column vector K_i is composed of dot products between x_i and all the training images, i.e., $K_i = (k(x_i,x_1),k(x_i,x_2),\ldots,k(x_i,x_N))^{\mathrm{T}}$, so K is a symmetrical matrix. m represents the mean of all the K_i, and μ is the mean of all the $\phi(x_i)$, i.e., $m = 1/N\sum_{i=1}^{N}K_i$ and $\mu = 1/N\sum_{i=1}^{N}\phi(x_i)$. For simplicity, $\phi(x_i)\in X_c$ means $x_i\in X_c$, and $K_i\in X_c$ means $x_i\in X_c$.

2.1. KPCA

Define matrix $\Phi(X) = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)]$ as the mapping matrix of X in F, and matrix $\bar{\Phi}(X) = [\mu, \mu, \ldots, \mu]$ is an N column matrix with vector μ. KPCA is equivalent to solving the problem of eigenvectors and eigenvalues of covariance matrix of $\Phi(X)$ [16]:

$$(W^{\phi})^{\mathrm{T}}[\Phi(X)-\bar{\Phi}(X)][\Phi(X)-\bar{\Phi}(X)]^{\mathrm{T}}W^{\phi} = \Lambda^{\phi}, \tag{1}$$

where W^{ϕ} is the matrix of eigenvectors, and Λ^{ϕ} is the diagonal matrix of eigenvalues in F.

Because W^{ϕ} is a linear transformation in F, any solution of eigenvector $w^{\phi}\in W^{\phi}$ can be represented by a combination of all the $\phi(x_i)$, $w^{\phi} = \sum_{i=1}^{N}\alpha_i\phi(x_i)$. Then we can rewrite $W^{\phi} = \Phi(X)\alpha$, where α is the matrix of coefficients. Since $\Phi(X)^{\mathrm{T}}\Phi(X) = K$ and $\Phi(X)^{T}\bar{\Phi}(X) = \bar{K}$, where matrix $\bar{K} = [m,m,\ldots,m]$ is an N column matrix with vector m; Eq. (1) can be rewritten as

$$\alpha^{\mathrm{T}}(K-\bar{K})(K-\bar{K})^{\mathrm{T}}\alpha = \Lambda^{\phi}. \tag{2}$$

Thus, the problem of KPCA is equivalent to solving the eigenvectors of the covariance matrix of K.

2.2. KDA

Define the between-class scatter S_{b}^{ϕ} and the within-class scatter S_{w}^{ϕ} in F as $S_{\mathrm{b}}^{\phi} = \sum_{i=1}^{C}N_i(\mu_i-\mu)(\mu_i-\mu)^{\mathrm{T}}$ and $S_{\mathrm{w}}^{\phi} = \sum_{i=1}^{C}\sum_{\phi(x_j)\in X_i}(\phi(x_j)-\mu_i)(\phi(x_j)-\mu_i)^{\mathrm{T}}$, where μ_i is the mean of the ith class samples in F. The idea of KDA is to perform LDA in F, i.e., maximizing the following objective function [25]:

$$J(W^{\phi}) = \arg\max_{W^{\phi}}\frac{|(W^{\phi})^{\mathrm{T}}S_{\mathrm{b}}^{\phi}W^{\phi}|}{|(W^{\phi})^{\mathrm{T}}S_{\mathrm{w}}^{\phi}W^{\phi}|}. \tag{3}$$

Similarly, any solution $w^{\phi}\in W^{\phi}$ can be represented by $w^{\phi} = \sum_{i=1}^{N}\alpha_i\phi(x_i)$ due to linear transform property, so Eq. (3) can be rewritten as

$$J(\alpha) = \arg\max_{\alpha}\frac{|\alpha^{\mathrm{T}}G_{\mathrm{b}}\alpha|}{|\alpha^{\mathrm{T}}G_{\mathrm{w}}\alpha|}, \tag{4}$$

where $G_{\mathrm{b}} = \sum_{i=1}^{C}N_i(m_i-\bar{m})(m_i-\bar{m})^{\mathrm{T}}$, $G_{\mathrm{w}} = \sum_{i=1}^{C}\Sigma_{K_j\in X_i}(K_j-m_i)(K_j-m_i)^{\mathrm{T}}$, $m_i = 1/N_i\sum_{j=1}^{N_i}K_j$ with $K_j\in X_i$. Thus, the problem of KDA is converted into finding the leading eigenvectors of $G_{\mathrm{w}}^{-1}G_{\mathrm{b}}$.

2.3. A new perception

From the above description, we see that KPCA is equivalent to solving the eigenvectors of covariance of K, and KDA is based on finding the leading eigenvectors of $G_{\mathrm{w}}^{-1}G_{\mathrm{b}}$, where G_{w} and G_{b} are the within-class and the between-class scatter based on K_i. So we can consider that the kernel-based subspace learning actually includes two independent steps: the kernel feature extraction with the dot product kernel and the linear subspace analysis on the kernel features. Dot products between an image and all the training images form the kernel feature vector of the image. $\{K_i\}$, $i = 1, 2, \ldots, N$, is the kernel feature set of the training images. KPCA and KDA are equivalent to performing PCA and LDA on the $\{K_i\}$, respectively. Since the dot product kernel is a nonlinear function of two images, using the linear subspace analysis on the kernel features one can get nonlinear features of the input images.

Based on this perception, it is easy to understand the projection of a new image z in KPCA or KDA subspace as $y = \alpha^{\mathrm{T}}K_z$, where $K_z = (k(z,x_1), k(z,x_2), \ldots, k(z,x_N))^{\mathrm{T}}$ is the kernel feature vector of the image z, i.e., its elements are dot products of the image z and all the training images. This is exactly the projection onto W^{ϕ} [16,25]. Thus, in a sense, we can regard the training images as a reference set. Before the kernel methods compare two images, they first measure the similarities between them and the training images, respectively. Fig. 1 gives an illustration. Given two samples y_i and y_j, their kernel features f_i and f_j are first extracted, respectively, and then their similarity $S(y_i,y_j)$ is measured in the PCA or LDA subspace that is obtained by the training samples.

It seems that this perception is also suitable for support vector machine (SVM) [14]. The discriminating function of SVM is: $f(x) = \mathrm{sgn}\{\sum_{i=1}^{n}\alpha_i y_i k(x,x_i)+b\}$, where x_i, $i = 1, 2, \ldots, n$ means support vector, and α_i is the non-zero weight of support vector x_i, and y_i represents the label {+1,−1} of x_i. In fact, the SVM classifier can be rewritten as $f(x) = \mathrm{sgn}\{\sum_{i=1}^{N}\alpha_i y_i k(x,x_i)+b\}$, where the

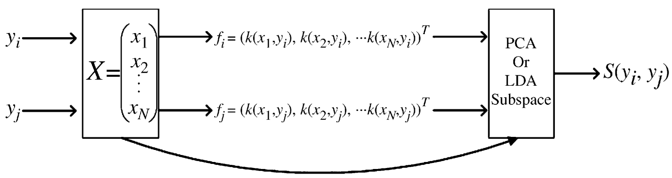

Fig. 1. Perception of KPCA and KDA.

weights α_i of the non-support vectors are equal to 0. If we let $\beta_i = \alpha_i y_i$, then $f(x) = \mathrm{sgn}(\beta^{\mathrm{T}} k_x + b)$, where $k_x = (k(x,x_1), k(x,x_2), \ldots, k(x,x_N))^{\mathrm{T}}$ is the kernel feature of the image x.

In addition, this perception helps to explain why some non-Mercer kernels can also achieve a good performance [8,20]. According to our perception, the kernel feature extraction is an independent and explicit step, and the kernel function can be regarded as a similarity function between two images [3]. Thus, we can take a similarity measure function as a valid kernel function if it is propitious to recognition, even though it does not satisfy the Mercer conditions. Moreover, based on this perception of explicit kernel feature extraction, we can further extend the kernel features.

3. A new extension of the kernel feature for visual recognition

In Section 2, we give a new perception of the kernel feature. Different from the idea of implicit feature space, the kernel feature in our perception becomes an explicit and independent step, so we can easily manipulate the kernel feature. In this section, we propose to extend the kernel feature vector of an image to a kernel feature matrix, and in order to efficiently deal with numerical computation a matrix-based KDA algorithm is adopted to learn discriminating features for visual recognition.

3.1. The kernel feature matrix

The traditional kernel methods often consider the global representations of the images in the form of vectors, and compute the dot products between an image and all the training images to form its kernel feature vector. Thus, they ignore detailed visual information except global information, and they cannot provide refined visual representation. Many studies show that global representations are sensitive to environment noise [4,20].

According to our perception, the kernel feature extraction is an explicit and independent step, so we can easily extend the kernel feature. It is known that, to better capture the complex structure and appearance of the objects, we should represent the images with different cues, such as texture, color, local components, and so on. Here, we propose to perform the dot product kernels on different cues, respectively, and the kernel feature vector is extended to a kernel feature matrix. This extension considers different information, respectively, so it should be more accurate for describing the images than the traditional one, only considering the global information. In a sense of dot product as similarity, this extension means using multiple similarities to measure two images, which is similar to human eyes that often tell us the similarities between two objects according to various cues.

Assuming that the images are represented by p cues, we use p dot product kernels on p cues, respectively. Then, the kernel feature vector K_i of the image x_i becomes a feature matrix $\tilde{K}_i$ as

$$\tilde{K}_i = \begin{bmatrix} k_1(x_i^1, x_1^1) & k_2(x_i^2, x_1^2) & \cdots & k_p(x_i^p, x_1^p) \\ k_1(x_i^1, x_2^1) & k_2(x_i^2, x_2^2) & \cdots & k_p(x_i^p, x_2^p) \\ \vdots & \vdots & \vdots & \vdots \\ k_1(x_i^1, x_N^1) & k_2(x_i^2, x_N^2) & \cdots & k_p(x_i^p, x_N^p) \end{bmatrix}, \tag{5}$$

where k_p is the dot production kernel function for the pth cue. We also call it multiple similarities-based kernel features.

This extended kernel feature embodies more information, but brings with it a problem of numerical computation, for the popular data analysis methods are all based on the vector-based data. The intuitive idea is to reshape $\tilde{K}_i$ as a vector with $N \times p$ elements. However, this reshaping loses coupled information existed in rows and columns of $\tilde{K}_i$, which is very useful for recognition. We can see that the rows of $\tilde{K}_i$ are multiple similarities information between two images, and the columns of $\tilde{K}_i$ represent the kernel features based on each cue. Moreover, the reshaping leads to expensive computation due to dimension increasing p times. For example, if there are 1000 training samples and 10 different cues, then the number of dimension becomes 10^4. With subspace learning, performing eigen decomposition in such a high-dimensional space may cause instability of numerical computation. In order to efficiently deal with these problems, we adopt the MKDA algorithm to learn nonlinear discriminating subspace with these matrix-based kernel features for recognition in the following.

3.2. Matrix-based KDA

Similar to the traditional KDA, based on the extracted kernel features, MKDA also employs the Fisher criterion that maximizes the between-class scatter and minimizes the within-class scatter, but it extends the vector-based norm into the Frobenius norm as in [26]. With the kernel features $\tilde{K}_i$, $i = 1, 2, \ldots, N$, the between-class scatter $\tilde{S}_{\mathrm{b}}$ and the within-class scatter $\tilde{S}_{\mathrm{w}}$ measured by the Frobenius norm are

$$\tilde{S}_{\mathrm{b}} = \sum_{i=1}^{C} N_i ||M_i - \bar{M}||_{\mathrm{F}}^2, \tag{6}$$

$$\tilde{S}_{\mathrm{w}} = \sum_{i=1}^{C} \sum_{\tilde{K}_j \in X_i} ||\tilde{K}_j - M_i||_{\mathrm{F}}^2, \tag{7}$$

where M is the mean matrix of all the $\tilde{K}_i$, and M_i represents the mean matrix of the ith class, and $\tilde{K}_i \in X_i$ means that $\tilde{K}_j$ belongs to the ith class.

The goal is to find the optimal projection matrices $L \in \Re^{N \times d_{\mathrm{L}}}$ and $R \in \Re^{p \times d_{\mathrm{R}}}$ which maximize $\tilde{S}_{\mathrm{b}}$ and minimize $\tilde{S}_{\mathrm{w}}$ in the low dimensional subspace of $L \otimes R$, i.e., maximizing $\tilde{S}_{\mathrm{b}}' = \sum_{i=1}^{C} N_i || L^{\mathrm{T}}(M_i - \bar{M})R||_{\mathrm{F}}^2$ and minimizing $\tilde{S}_{\mathrm{w}}' = \sum_{i=1}^{C} \sum_{\tilde{K}_j \in X_i} ||L^{\mathrm{T}}(\tilde{K}_j - M_i)R||_{\mathrm{F}}^2$ at the same time.

Because $||X||_{\mathrm{F}}^2 = \mathrm{trace}(XX^{\mathrm{T}})$, $\tilde{S}_{\mathrm{b}}'$ and $\tilde{S}_{\mathrm{w}}'$ can be written as $\tilde{S}_{\mathrm{b}}' = \mathrm{trace}(L^{\mathrm{T}} D_{\mathrm{b}}^{\mathrm{R}} L)$ and $\tilde{S}_{\mathrm{w}}' = \mathrm{trace}(L^{\mathrm{T}} D_{\mathrm{w}}^{\mathrm{R}} L)$ when R is given, where

$$D_{\mathrm{b}}^{\mathrm{R}} = \sum_{i=1}^{C} N_i (M_i - \bar{M}) R R^{\mathrm{T}} (M_i - \bar{M})^{\mathrm{T}}, \tag{8}$$

$$D_{\mathrm{w}}^{\mathrm{R}} = \sum_{i=1}^{C} \sum_{\tilde{K}_j \in X_i} (\tilde{K}_j - M_i) R R^{\mathrm{T}} (\tilde{K}_j - M_i)^{\mathrm{T}}. \tag{9}$$

Then we can get the optimal projection L by maximizing $\mathrm{trace}((L^{\mathrm{T}} D_{\mathrm{w}}^{\mathrm{R}} L)^{-1} (L^{\mathrm{T}} D_{\mathrm{b}}^{\mathrm{R}} L))$, i.e., computing the eigenvectors of $(D_{\mathrm{w}}^{\mathrm{R}})^{-1} D_{\mathrm{b}}^{\mathrm{R}}$.

Similarly, if L is fixed, we can rewrite $\tilde{S}_{\mathrm{b}}'$ and $\tilde{S}_{\mathrm{w}}'$ as $\tilde{S}_{\mathrm{b}}' = \mathrm{trace}(R^{\mathrm{T}} D_{\mathrm{b}}^{\mathrm{L}} R)$ and $\tilde{S}_{\mathrm{w}}' = \mathrm{trace}(R^{\mathrm{T}} D_{\mathrm{w}}^{\mathrm{L}} R)$, because of $\mathrm{trace}(AB) = \mathrm{trace}(BA)$, where

$$D_{\mathrm{b}}^{\mathrm{L}} = \sum_{i=1}^{C} N_i (M_i - \bar{M})^{\mathrm{T}} L L^{\mathrm{T}} (M_i - \bar{M}), \tag{10}$$

Table 1
The MKDA algorithm

Input: $\tilde{K}_1, \tilde{K}_2, \ldots, \tilde{K}_N$
Initialization: Set $R_0 \leftarrow (I_{d_R}, 0)^{\mathrm{T}}$, and compute the mean M_i of the i-th class for each i, and the global mean M.
For $t = 1$ to T
(1) For a given R_{t-1}, compute $D_{\mathrm{w}}^{\mathrm{R}}$ and $D_{\mathrm{b}}^{\mathrm{R}}$ using Eqs. (8) and (9), and get the optimal L_t by solving the first d_{L} leading eigenvectors of $(D_{\mathrm{w}}^{\mathrm{R}})^{-1}D_{\mathrm{b}}^{\mathrm{R}}$.
(2) Based on L_t, compute $D_{\mathrm{w}}^{\mathrm{L}}$ and $D_{\mathrm{b}}^{\mathrm{L}}$ as in Eqs. (10) and (11), and get the optimal R_t by solving the first d_{R} leading eigenvectors of $(D_{\mathrm{w}}^{\mathrm{L}})^{-1}D_{\mathrm{b}}^{\mathrm{L}}$.
(3) If $t>1, \|L_t - L_{t-1}\| < \varepsilon$ and $\|R_t - R_{t-1}\| < \varepsilon$, break; else, continue.
End
Output: $L = L_t$ and $R = R_t$

$$D_{\mathrm{w}}^{\mathrm{L}} = \sum_{i=1}^{C} \sum_{\tilde{K}_j \in X_i} (\tilde{K}_j - M_i)^{\mathrm{T}} L L^{\mathrm{T}} (\tilde{K}_j - M_i). \tag{11}$$

Then the optimal projection R can be obtained by maximizing $\mathrm{trace}((R^{\mathrm{T}} D_{\mathrm{w}}^{\mathrm{L}} R)^{-1}(R^{\mathrm{T}} D_{\mathrm{b}}^{\mathrm{L}} R))$, i.e., solving the eigenvectors of $(D_{\mathrm{w}}^{\mathrm{L}})^{-1}D_{\mathrm{b}}^{\mathrm{L}}$.

Thus, the final optimal solution can be computed by an iterative procedure, shown in Table 1. It can be found that the MKDA algorithm not only avoids the eigen-decomposition in the $N \times p$ dimensional space, but also well preserves the geometric relations of row and column of $\tilde{K}_i$. In addition, in the traditional KDA, the available dimension has the upper bound $C-1$, while MKDA has no such constraint.

For a new pattern z, its projection is: $Y_i = L^{\mathrm{T}} \tilde{K}_z R$, where $\tilde{K}_z$ is the kernel feature matrix of the image z, $\tilde{K}_z = \{A_{i,j} = \{k_i(z^i, x_j^i)\}\}$, $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, N$.

4. Experiments

Our experiments are conducted on the FERET face database [15], the COIL-100 object database [13], and the Wang's nature image database [2]. We compare the proposed MKDA with KDA, LDA, and the method of reshaping the kernel feature matrix $\tilde{K}_i$ as a vector (we denote it as VKDA for simplicity). The Gaussian kernel is employed for all the k_p in MKDA, VKDA, and KDA. As for the parameter γ, we set it as $\gamma = \beta/s$, where s is the dimension of x and y, and β is an adjustable constant. It seems that $\beta = 0.5$ is better for KDA among [0.1 1] on the three databases. For comparison and simplicity, we simply set $\beta = 0.5$ in our experiments for all the kernel functions. The nearest neighbor classifier is used for classification.

4.1. The FERET database

The FERET database is widely used to evaluate the performance of face recognition methods [15]. Our experimental data include FA and FB sets, and 1000 front view face images selected from training CD of the FERET database. There are 1196 images in FA set and 1195 images in FB set, and all of the subjects have only one image in FA and FB sets. We use 1000 images from the training CD as the training set. FA images are used for gallery images, and FB images are taken as probe images. All the images are cropped to 48×54 by fixing two eye locations at (12,14) and (36,14). The variations include illumination, expression and tiny pose changes.

In this group experiment, we use Gabor representations of the image, for Gabor-based face recognition has achieved great success [7,18] due to the good properties of Gabor filters, such as spatial localization, spatial frequency characteristic, and orientation selectivity. Similar to previous studies, 40 Gabor filters are adopted, i.e., five scales and eight orientations, and then 40 Gabor images are obtained for each image. In previous works, such as LDA [7] and KDA [18], they often consider 40 Gabor images together and reshape them into a vector. To reduce the computation complexity, the popular technique is to downsample the Gabor images. Actually 40 Gabor images correspond to different scales and orientations, respectively, so they should have different response characteristics. Here, we take them as 40 different cues of the face image in MKDA, i.e., $p = 40$. Similar to [7,18], we downsample the Gabor images with sampling factor $\rho = 4$ and reshape them as a vector to test LDA and KDA methods.

Table 2
The best recognition rates of the four methods

Method	LDA	KDA	VKDA	MKDA
Recognition rate (%)	94.48	95.07	95.99	97.57

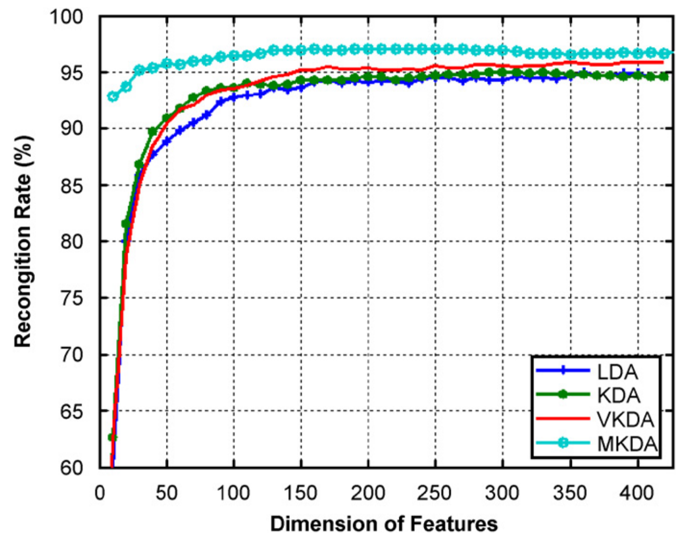

Fig. 2. Recognition rates with different dimensions of the four methods.

Table 2 reports the best recognition rates of the four methods. MKDA gives a higher recognition rate than LDA, KDA and VKDA. The performance of MKDA is also comparable with the result reported in the latest literature [17]. MKDA gets the recognition rate of 97.57%, while the best recognition rate of [17] is 97.2%. We also investigate the recognition performance with variation of feature numbers shown in Fig. 2, where we fix the feature numbers $d_{\mathrm{R}} = 16$ in the right matrix R, and test the performance of MKDA with the variation of left matrix L. We can see the superiority of MKDA. Fig. 3 shows the performance of MKDA with variation of d_{L} and d_{R}. We can see that the performance of MKDA stabilizes near the best result when $d_{\mathrm{R}} \geqslant 10$ and d_{L} between 100 and 600. Comparing with the original dimension of kernel features, $10^3 \times 40$, MKDA can get a good performance with a small number of dimensions.

4.2. The COIL-100 database

The COIL-100 [13] database has 100 different objects, and each object has 72 color images at pose intervals of 5°. The image size is 128×128. In our experiments, we convert the images into gray images, and evaluate the robustness to such view changes for

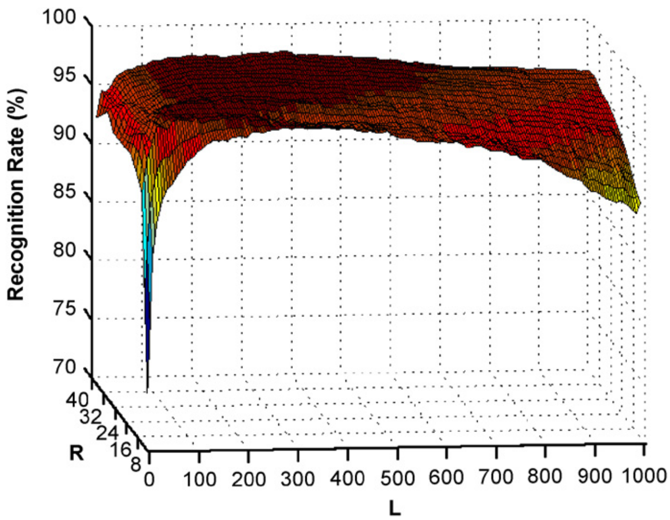

Fig. 3. The performance of MKDA with different dimensions.

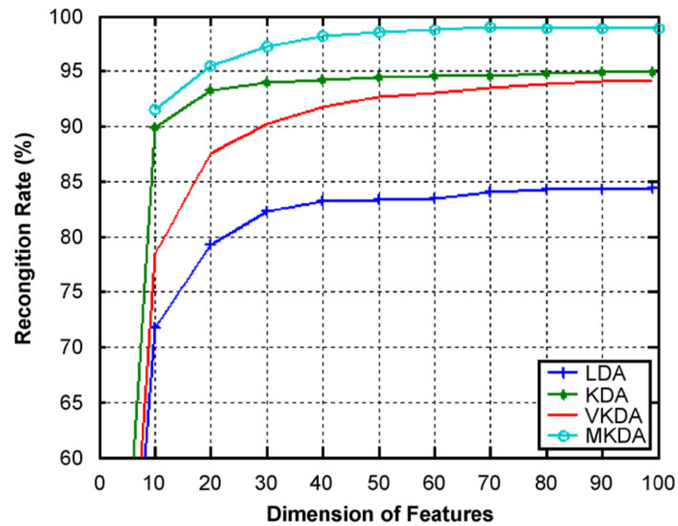

Fig. 4. Recognition rates with different dimensions of the four methods.

Table 3
The best recognition rates of the four methods

Method	LDA	KDA	VKDA	MKDA
Recognition rate (%)	84.59	95.65	94.24	99.22

Table 4
The classification errors of four methods

Method	LDA	KDA	VKDA	MKDA
Error rate (%)	12	10.9	13.5	10.4

general objects. Same as in [9,10], we select 18 views of each of the 100 objects as the training samples, starting with the pose at 0° and continuing at intervals of 20°, and the rest of the images for testing. For simplicity, we divide the images into 16 patches with size of 32×32, and take these patches as local components of the object to test MKDA, i.e., $p = 16$. Thus, the dimension of a kernel feature matrix is 1800×16.

The best recognition rates of the four methods are reported in Table 3. The results are similar to the above results on the FERET database, i.e., MKDA outperforms the other methods. The recognition rate of MKDA is also comparable with the reported results in [9,10]. Using the same testing protocol, methods in the literature report the recognition rate from 87.5% to 99.9%. We simply divide the image into 16 patches on the average, and take them as local components in the experiments. Perhaps the performance of MKDA can be further improved if we automatically detect local components according to the object structure. Fig. 4 gives the recognition performances of four methods with the variation of feature numbers. Because the class numbers of the training images are 100, the dimensions of LDA, KDA and LKDA are equal to 99. As for MKDA, we set $d_R = 8$ and investigate d_L from 10 to 100. The performances are similar to those on the FERET database too.

4.3. The Wang's database

The Wang's database consists of 1000 nature images of 10 categories, each represented by 100 images, illustrating the following themes [2,9]: African people and villages, beach, buildings, buses, dinosaurs, elephants, flowers, horses, mountain and glaciers, and food. Such common categories exhibit high intra-class variability. The images are of size 384×256. Same as in [2,9], the leave-one-out testing protocol is adopted to test the proposed method for image classification.

Four kinds of visual features are used to represent the images [21,22]: color histogram, color moments, wavelet-based texture and directionality. A color histogram is taken in HSV space with quantization of $8 \times 4 = 32$ bins on H and S channels; the first three moments from each of the three color channels are used for color moment; a 24-dimensional PWT-based wavelet texture features and an 8-dimensional directionality features are contained to construct an 73-dimensional feature vector for each image. For MKDA, these four kinds of visual features are considered, respectively, i.e., $p = 4$. Thus, the dimension of the kernel feature matrix is 999×4. Table 4 reports the error rates of four methods, where MKDA keeps $d_L = 100$ and $d_R = 3$. Same as in the above experiments, MKDA has a better classification performance than the three related methods. The error rate of MKDA is 10.4%. This result is also better than the published results. Error rates in the literature vary from 62.5% to 15.9% [2,9].

5. Discussions

The above experiments on the three different databases show that MKDA outperforms LDA, KDA and VKDA. It shows the advantage of the extension of the kernel feature and the matrix-based KDA algorithm. We can see that simply reshaping the kernel feature matrix as a vector loses the coupled information in the kernel feature matrix, and the high-dimensional vector also causes instability of numerical computation due to performing eigen-decomposition in a higher-dimensional space.

The proposed MKDA is based on the new perception that the kernel method is comprised of an explicit feature extraction step and a feature analysis step. According to the principle of the implicit feature space, if we want to use multiple cues together, we should map each of them into the implicit feature space, respectively. However, the implicit feature space is unknown to us, so it is difficult to integrate different implicit feature space together. Based on our perception of the kernel feature, it is easy to understand the extended kernel matrix features in MKDA. From the sense of similarity, the kernel matrix feature embodies the property of multiple similarity measurement of human vision system. In addition, this perception also relaxes the criterion of kernel selection, and helps to interpret some useful non-Mercer kernels [8,20]. In this paper, we only adopt the Gaussian kernel for all the cues in MKDA, and its parameter is set in terms of KDA. The performance of MKDA can be further improved if we can design a special kernel for each cue. Besides the kernel selection, this new perception brings out another interesting issue. The kernel feature of an image is comprised of the dot products between the image and all the training samples. In a sense, we can regard the training samples as a reference set, and the similarity between two images is actually a relative similarity with respect to the reference set. In the traditional kernel methods including the proposed method, the reference set is the training set. How to select an effective reference set is a very interesting issue, which is worthy of further investigation in future work.

6. Conclusions

In this paper, we first give a new perception of the kernel methods. The kernel methods are actually divided into two separated steps, i.e., the kernel feature extraction and the data analysis based on the extracted kernel features. Dot products between an image and all the training images form its kernel features. Based on this perception, we extend the kernel feature vector to a kernel feature matrix, because the images are often represented by different cues, such as color, texture, local components, and so on. In order to efficiently deal with the problem of numerical computation, a matrix-based KDA algorithm is developed to learn discriminating kernel features for visual recognition. Experiments on the FERET face database, the COIL-100 object database, and the Wang's nature image database show the advantage of the proposed method.

Acknowledgment

This work is supported by the projects of Nature Sciences Foundation of China under Grant no. 60405005, 60675003, and 60605004.

Reference

[1] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[2] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, J. Mach. Learn. Res. 5 (2004) 913–939.
[3] N. Cristianini, J.S. Taylor, J. Kandola, Spectral kernel methods for clustering, in: Proceedings of the International Conference on Neural Information Processing Systems, 2001.
[4] B. Heisele, P. Ho, T. Poggio, Face recognition with support vector machines: global verus component-based approach, in: Proceedings of the International Conference on Computer Vision, 2001.
[5] Q.S. Liu, R. Huang, H.Q. Lu, S.D. Ma, Face recognition using kernel-based Fisher discriminant analysis, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2002.
[6] Q.S. Liu, H.Q. Lu, S.D. Ma, Improving kernel Fisher discriminant analysis for face recognition, IEEE Trans. Circuit Syst. Video Technol. 14 (1) (2004) 42–49.
[7] C.J. Liu, H. Wechsler, Gabor feature-based classification using the enhanced Fisher linear discriminant model for face recognition, IEEE Trans. Image Process. 11 (4) (2002) 467–476.
[8] S. Lyu, Mercer kernels for object recognition with local features, in: Proceedings of the International on Conference Computer Vision and Pattern Recognition, 2005.
[9] R. Maree, P. Geurts, J. Piater, L. Wehenkel, Random sub-windows for robust image classification, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2005.
[10] J. Matas, S. Obdrzalek, Object recognition methods based on transformation covariant features, in: Proceedings of the European on Conference Signal Processing, 2004.
[11] S. Mika, G. Ratsch, J. Weston, Fisher discriminant analysis with kernels, in: Proceedings of the Neural Networks for Signal Processing Workshop, 1999.
[12] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 696–710.
[13] H. Murase, S.K. Nayar, Visual learning and recognition of 3D objects from appearance, Int. J. Comput. Vision 14 (1) (1995) 5–24.
[14] E. Osuna, R. Freund, F. Girosi, Support vector machines: training and applications, Tech Report, AI Lab, MIT, 1997.
[15] P.J. Phillips, H. Wechsler, J. Huang, P. Rauss, The FERET database and evaluation procedure for face recognition algorithms, Image Vision Comput. 16 (5) (1998) 295–306.
[16] B. Scholkopf, A. Smola, K.R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (5) (1998) 1299–1319.
[17] S.G. Shan, P. Yang, X.L. Chen, W. Gao, AdaBoost Gabor Fisher classifier for face recognition, in: Proceedings of the International Workshop on Analysis and Modeling of Faces and Gestures, joint with ICCV, 2005.
[18] L. Shen, L. Bai, Gabor feature-based face recognition using kernel methods, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2004.
[19] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognitive Neurosci. 3 (1) (1991) 72–86.
[20] N. Vasconcelos, P. Ho, P. Moreno, The Kullback–Leibler kernel as a framework for discriminant and localized representations for visual recognition, in: Proceedings of the European Conference on Computer Vision, 2004.
[21] H. Wu, H.Q. Lu, S.D. Ma, A practical SVM-based algorithm for ordinal regression in image retrieval, in: Proceedings of the ACM Multimedia, 2003.
[22] W. Yan, Q.S. Liu, H.Q. Lu, S.D. Ma, Multiple similarities-based kernel subspace learning for image classification, in: Proceedings of the Asian Conference on Computer Vision, 2006.
[23] S. Yan, D. Xu, L. Zhang, B. Zhang, H. Zhang, Coupled kernel-based subspace learning, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, 2005.
[24] M.H. Yang, N. Ahuja, D. Kriegman, Face recognition using kernel eigenfaces, in: Proceedings of the International Conference on Image Processing, 2000.
[25] M.H. Yang, Kernel eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2002.
[26] J.P. Ye, R. Janardan, Q. Li, Two-dimensional linear discriminant analysis, in: Proceedings of the International Conference on Neural Information Processing Systems, 2004.
[27] W. Zhao, R. Chellappa, P.J. Phillips, Subspace linear discriminant analysis for face recognition, Tech. Report CAR-TR-914, University of Maryland, 1999.

Qingshan Liu received MS in Auto Control from South-East University in 2000. He got the Ph.D. degrees in Pattern Recognition and Intelligence System from the National Laboratory of Pattern Recognition, CAS in 2003, where he is an associate professor now. He received the president scholarship of the Chinese Academy of Sciences in 2003. From June 2004 to April 2005, he worked as an associate researcher at the Multimedia Laboratory in Chinese University of Hong Kong. His interests include Face Analysis, Image & Video Analysis, Machine Learning, computer vision, etc. He is a member of the IEEE.

Hongliang Jin received the Ph.D. degrees in Pattern Recognition and Intelligence System from the National Laboratory of Pattern Recognition, CAS in 2006. Now he is a researcher at Panasonic China Research Center.

Xiaoou Tang received the B.S. degree from the University of Science and Technology of China, Hefei, in 1990, and the M.S. degree from the University of Rochester, Rochester, NY, in 1991. He received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, in 1996. He is a professor and the director of Multimedia Lab in the Department of Information Engineering, the Chinese University of Hong Kong. He is also the group manager of the Visual Computing Group at the Microsoft Research Asia. He is a local chair of the IEEE International Conference on Computer Vision (ICCV) 2005, an area chair of ICCV'07, a program chair of ICCV'09, and a general chair of the ICCV International Workshop on Analysis and Modeling of Faces and Gestures 2005. He is a guest editor of the Special Issue on Underwater Image and Video Processing for

IEEE Journal of Oceanic Engineering and the Special Issue on Image- and Video-based Biometrics for IEEE Transactions on Circuits and Systems for Video Technology. He is an associate editor of IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI). His research interests include computer vision, pattern recognition, and video processing.

Hanqing Lu received his B.E. degree and M.S. degree both from Harbin Institute of Technology in 1982 and 1985, respectively. He received his Ph.D. degree from Huazhong University of Science and Technology in 1992. Since 1992, he has been with the National Laboratory of Pattern Recognition, the Institute of Automation of Chinese Academy of Sciences, where he is now a professor. Since 2001, he was made vice head in the Laboratory. His research interests include image and video analysis, content-based image and video retrieval, and pattern recognition.

Songde Ma received the B.S. degree from Tsinghua University in 1968. He received the Ph.D. degree in 1983 and the "Doctorat d'Etat es Science" degree in 1986 from University of Paris 6. He was an invited researcher in the computer vision laboratory of University of Maryland, College Park, USA in 1983. He was a researcher in the robot vision laboratory in INRIA, France in 1984–1986. Since 1986, he has been a research professor in the National Pattern Recognition Laboratory of the Institute of Automation, Chinese Academy of Sciences. He was the president of the Institute of Automation of Chinese Academy of Sciences in 1996–2000. He has been the vice-minister of the Ministry of Science and Technology (MOST) of China since April 2000. He is a senior member of IEEE. His research interests include computer vision, computer graphics, robotics, neural computing.