Neural Process Lett (2010) 31:1–15
DOI 10.1007/s11063-009-9123-3

A New Canonical Correlation Analysis Algorithm with Local Discrimination

Yan Peng · Daoqiang Zhang · Jianchun Zhang

Published online: 11 November 2009

Abstract In this paper, a new feature extraction algorithm is developed based on canonical correlation analysis (CCA), called Local Discrimination CCA (LDCCA). The method considers a combination of local properties and discrimination between different classes. Not only the correlations between sample pairs but also the correlations between samples and their local neighborhoods are taken into consideration in LDCCA. Effective class separation is achieved by maximizing local within-class correlations and minimizing local between-class correlations simultaneously. Besides, a kernel version of LDCCA (KLDCCA) is proposed to cope with nonlinear problems in experiments. The experimental results on an artificial dataset, multiple feature databases and face databases including ORL, Yale, AR validate the effectiveness of the proposed methods.

Keywords Canonical correlation analysis · Feature extraction · Local discrimination · Dimensionality reduction

1 Introduction

Canonical correlation analysis (CCA [1]), just like Principal component analysis (PCA [2]), is an effective feature extraction method for dimensionality reduction and data visualization. PCA is a single-modal method, which deals with data samples obtained from a single information channel or view. In contrast, CCA is typically used for multi-view data samples, which are obtained either from various information sources, e.g. sound and image, or from

Y. Peng · D. Zhang (✉) · J. Zhang
Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, 210016 Nanjing, China
e-mail: dqzhang@nuaa.edu.cn

Y. Peng
e-mail: pengy@nuaa.edu.cn

J. Zhang
e-mail: jczhang@nuaa.edu.cn

different features measured from a single source. In the past decades, CCA and its variants have been successfully used in many research areas such as facial expression recognition [3], image analysis [4], position estimation of robots [5], parameter estimation of posture [6], data regression analysis [7], image texture analysis [8], image retrieval [9], content based text mining [10] and asymptotic convergence of the functions [11]. Given a data set with two views X and Y, the goal of CCA is to seek a set of basis vector pairs which would maximize the correlation of the two views when been projected into lower dimensional space.

CCA is a linear dimensionality reduction method. However, there are many nonlinear relationships between features in practice. There will be under-fitting phenomenon when learning with CCA under a nonlinear circumstance. To solve the problem, several approaches have been proposed, e.g. kernel based methods [6,12], approaches with neural networks [13], and locality based ones [14–17]. With the help of well-known “kernel trick”, KCCA first maps the data into higher dimensional space (referred as feature space) through implicit nonlinear mappings: $\phi : x \rightarrow \phi(x)$ and $\varphi : y \rightarrow \varphi(y)$, then linear CCA is performed in the feature space [6,12]. Thus, a nonlinear problem in the original space is converted into a linear one in the feature space by doing this. KCCA helps to reveal the nonlinear relationships hidden behind original data. However, like many other kernel methods, one disadvantage of KCCA is the choice of appropriate kernels and kernel parameters [8,16], which is still an ‘open problem’. Neural networks based nonlinear CCA suffers from some intrinsic limitations such as long-time training, slow convergence and local minima [13].

In recent years, locality-preserving methods have achieved a remarkable flourish in dimensionality reduction research. Typical methods include local principal component analysis [14], locally linear embedding (LLE) [15], Isomap [16], and locality preserving projection (LPP [17]), etc. All above locality based approaches share such a character that they preserve the local structure information in original data and thus can discover the low dimensional manifold structure embedded in the original high dimensional space. More recently, locality preserving CCA (LPCCA) is proposed to incorporate local structure information into CCA [18]. LPCCA investigates canonical correlation problem in a small neighborhood by decomposing the global nonlinear problem into many local linear ones first, then getting the sum of these sub-problems. Consequently, in each small neighborhood field the problem can be solved by linear CCA and the global problem then could be solved by optimizing the combination of these local sub-problems. Nevertheless, CCA and LPCCA only concern the correlation between sample pairs and they are not designed to utilize the class information of samples that is essential for classification. More recently, Farquhar et al. [12] propose SVM-2K, which combines KCCA and the support vector machine (SVM) classifier into a single optimization problem and thus guarantees that the directions obtained by KCCA will be best suited to the classification task. Besides, Sun et al. [28] propose discriminant CCA and performed supervised feature extracting through combining two view by CCA for multimodal recognition. In this paper, we propose a new CCA model with local discrimination (called LDCCA), which introduces the class information of samples to classical CCA and consider the local correlations of the within-class sets and the between-class sets. Moreover, we extend LDCCA to kernel version (KLDCCA) to extract nonlinear features effectively. We expect that features extracted by LDCCA and KLDCCA would maximize the within-class correlation and minimize between-class correlation simultaneously.

The rest of this paper is organized as follows. In Sect. 2, linear CCA is briefly described. Section 3 derives the proposed LDCCA and gives the corresponding pseudo-code description. In Sect. 4, we present the generalization of LDCCA through kernelization. The experimental results are given in Sect. 5. At last, we conclude this paper in Sect. 6.

2 Canonical Correlation Analysis

Given a set of pair-wise samples $\{(x_i, y_i)\}_{i=1}^n \in \mathcal{R}^p \times \mathcal{R}^q$, where $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ are obtained from different information channels. Suppose that $X = [x_1, \ldots, x_n] \in \mathbb{R}^{p\times n}$ and $Y = [y_1, \ldots, y_n] \in \mathbb{R}^{q\times n}$, the aim of CCA is to find two sets of basis vectors $\omega_x \in \mathrm{R}^p$ and $\omega_y \in \mathrm{R}^q$ for X and Y, respectively to maximize the correlation coefficient between $\omega_x^T X$ and $\omega_y^T Y$. The process is formularized as follows:

$$\rho = \max_{\omega_x, \omega_y} \frac{\omega_x^T C_{xy} \omega_y}{\sqrt{\left(\omega_x^T C_{xx} \omega_x\right)\left(\omega_y^T C_{yy} \omega_y\right)}} \tag{1}$$

where $C_{xx} = E\left[xx^T\right] = XX^T$, $C_{yy} = E\left[yy^T\right] = YY^T$, and $C_{xy} = E\left[xy^T\right] = XY^T$. The solution of CCA can be obtained by computing a generalized eigenvalue decomposition problem. More details about derivation and solution of CCA can be found in [9].

3 Local Discriminative Canonical Correlation Analysis

In order to improve the performance of CCA in classification tasks, we incorporate the idea of local discriminant analysis into CCA, which is referred to as LDCCA. The standard CCA optimization problem is slightly modified so that the cross-covariance matrix (between the two views) C_{xy} in Eq. 1 is replaced by a term $\tilde{C}_{xy}$ which takes class information into account. It is defined as the sum of local within class covariance matrices penalized by the sum of local between class covariance matrices. It implies that samples nearby in the original space should be close together in the feature space. The idea of locality is defined in terms of k nearest neighborhood (k-NN).

Given n pairs of samples $\{(x_i, y_i)\}_{i=1}^n \in \mathcal{R}^p \times \mathcal{R}^q$, the aim of LDCCA is to find a set of directions $\omega_x \in \mathcal{R}^p$ and $\omega_x \in \mathcal{R}^q$ to maximize the correlation coefficient of within-class k-NN sample features and minimize the correlation of between-class k-NN sample features. The objective function of LDCCA can be formularized as follows:

$$\rho = \max \frac{\omega_x^T C_{xy} \omega_y}{\sqrt{\left(\omega_x^T \tilde{C}_{xx} \omega_x\right)\left(\omega_y^T C_{yy} \omega_y\right)}} \tag{2}$$

where $\tilde{C}_{xy} = C_\omega - \eta C_b$. C_ω denotes local within-class covariance matrix and C_b denotes local between-class covariance matrix, η is a balancing factor, which makes a trade-off between C_ω and C_b. C_ω and C_b are defined as:

$$\begin{aligned} C_\omega &= \sum_{i=1}^n \sum_{x_k \in \mathcal{N}^I(x_i), y_k \in \mathcal{N}^I(y_i)} x_i y_k^T + x_k y_i^T \\ C_b &= \sum_{i=1}^n \sum_{x_k \in \mathcal{N}^E(x_i), y_k \in \mathcal{N}^E(y_i)} x_i y_k^T + x_k y_i^T \end{aligned} \tag{3}$$

Here, $\mathcal{N}^I(x_i)$ denotes within-class k nearest neighborhoods of $x_{i,}$, $\mathcal{N}^E(x_i)$ denotes between-class k nearest neighborhoods of $x_{i,}$. In other words, $\mathcal{N}^I(x_i)$ represents the set of points which are the most similar with $x_{i,}$ in the same class, while $\mathcal{N}^E(x_i)$ represents the set of points which are the most similar with $x_{i,}$ in different classes. We use the standard Euclidean distance to

Table 1 The Algorithm of LDCCA

Input:	Training data matrix $X = [x_1 \dots x_n] \in \mathcal{R}^{p\times n}$, $Y = [y_1 \dots y_n] \in \mathcal{R}^{q\times n}$, parameters k, η and d
Output:	Projection vectors $W_x = \left[\omega_{x1} \dots \omega_{xd}\right]$ and $W_y = \left[\omega_{y1} \dots \omega_{yd}\right]$
Step 1:	Compute local covariance matrices C_ω and C_b according to Eq. (3)
Step 2:	Get covariance matrices $\tilde{C}_{xy} = C_\omega - \eta C_b, C_{xx} = XX^T, C_{yy} = YY^T$;
Step 3:	Compute matrix $H = C_{xx}^{-1/2} \tilde{C}_{xy} C_{yy}^{-1/2}$;
Step 4:	Perform SVD decomposition $H = UDV^T$;
Step 5:	Choose $\left[U_1 \dots U_d\right]$ and $\left[V_1 \dots V_d\right], d < n$;
Step 6:	Obtain $W_x = C_{xx}^{-1/2}\left[U_1 \dots U_d\right], W_y = C_{yy}^{-1/2}\left[V_1 \dots V_d\right]$;

measure the similarity between data points. Note that both x and y have been centralized. Similar to CCA, the solution of LDCCA can also be obtained by computing a generalized engenvalue decomposition problem, and “Appendix” gives the detailed derivation of the solution. The pseudo-code of LDCCA algorithm is summarized in Table 1.

After obtaining eigenvectors W_x, W_y corresponding to d generalized eigenvalues λ_i, $i = 1 \dots d$. For any sample $x_{i,} y_i$, we can extract features as follows:

$$\begin{aligned} &(a)\, W_x^T x + W_y^T Y \\ &(b) \begin{bmatrix} W_x^T & x \\ W_y^T & y \end{bmatrix} \end{aligned} \tag{4}$$

where $W_x = [\omega_{x1} \dots \omega_{xd}] \in R^{p\times d}$, $W_y = \left[\omega_{y1} \dots \omega_{yd}\right] \in R^{q\times d}, d < \min(p, q)$ The two feature combination methods in Eq.(4) are referred to as parallel combination and serial combination [19], denoted as PR1 and PR2, respectively, throughout this paper. With the fused features, we can classify them using any classifier. In this paper, we use the nearest neighbor classifier.

4 A Generalization of LDCCA via Kernelization

Kernelization is an effective method when processing nonlinear problems, and various kernel based learning methods, such as KCCA [6], KPCA [20], KICA [21], have been proposed. Kernel methods enable us to work within higher dimensional feature spaces by defining weight vectors implicitly as linear combinations of the training examples. For example, when using the Gaussian kernel, this even makes it practical to learn in infinite dimensional spaces. There are also other kernels on a range of different data types such as the string kernels for text, graph kernels for graphs [22]. In this section, we will extend LDCCA to its kernel version (KLDCCA).

Given n samples $\{(x_i, y_i)\}_{i=1}^n \in \mathcal{R}^p \times \mathcal{R}^q$, let $X = [x_1 \dots x_n]$ and $Y = [y_1 \dots y_n]$, and we suppose that all samples have been centralized. The solution vector of LDCCA can be expressed as $\omega_x = X\alpha$ and $\omega_y = Y\beta$, where the components of α and β denote linear combination coefficients. Assume the nonlinear mapping $\phi : x \to \phi(x)$ and $\varphi : y \to \varphi(y)$, which map samples into feature space, let $\phi(X) = [\phi(x_1) \dots \phi(x_n)]$ and $\varphi(Y) = [\varphi(y_1) \dots \varphi(y_n)]$, which are data matrices in feature space. We assume that the sample means are equal to zeros for simplicity, i.e. $\phi(x) = \frac{1}{n}\sum_{i=1}^n \phi(x_i) = 0$ and $\varphi(y) = \frac{1}{n}\sum_{i=1}^n \varphi(y_i) = 0$. Because KLDCCA is in essence performing LDCCA in feature space, the solution vector of

KLDCCA ω_ϕ and ω_ϕ can be expressed by linear combination of samples $\{\phi(x_i)\}_{i=1}^n$ and $\{\varphi(y_i)\}_{i=1}^n$, i.e. $\omega_\phi = \phi(X)\alpha$ and $\omega_\varphi = \varphi(Y)\beta$. The objective function of KLDCCA is to optimize

$$\rho = \max_{\omega_x, \omega_y} \frac{\alpha^T K_{xy} \beta}{\sqrt{\left(\alpha^T K_{xx} \alpha\right)\left(\beta^T K_{yy} \beta\right)}} \tag{5}$$

where,

$$\begin{aligned} K_{xx} &= \left(\phi(X)^T \phi(X)\right)\left(\phi(X)^T \phi(X)\right) = K_x K_x \\ K_{yy} &= \left(\varphi(Y)^T \varphi(Y)\right)\left(\varphi(Y)^T \varphi(Y)\right) = K_y K_y \end{aligned} \tag{6}$$

$K_{xy} = K_\omega - \eta K_b$, η is a balancing factor, K_ω is local within-class kernel matrix and K_b is local between-class kernel matrix which are defined as follows:

$$\begin{aligned} K_\omega &= \sum_{i=1}^{n} \sum_{x_k \in \mathcal{N}^I(x_i), y_k \in \mathcal{N}^I(y_i)} \phi(x_i)^T \phi(x_i) \varphi(y_k)^T \varphi(y_i) + \phi(x_i)^T \phi(x_k) \varphi(y_i)^T \varphi(y_i) \\ &= \sum_{i=1}^{n} \sum_{x_k \in \mathcal{N}^I(x_i), y_k \in \mathcal{N}^I(y_i)} (K_x)_{ii} \left(K_y\right)_{ki} + (K_x)_{ik} \left(K_y\right)_{ii} \\ K_b &= \sum_{i=1}^{n} \sum_{x_k \in \mathcal{N}^E(x_i), y_k \in \mathcal{N}^E(y_i)} \phi(x_i)^T \phi(x_i) \varphi(y_k)^T \varphi(y_i) + \phi(x_i)^T \phi(x_k) \varphi(y_i)^T \varphi(y_i) \\ &= \sum_{i=1}^{n} \sum_{x_k \in \mathcal{N}^E(x_i), y_k \in \mathcal{N}^E(y_i)} (K_x)_{ii} \left(K_y\right)_{ki} + (K_x)_{ik} \left(K_y\right)_{ii} \end{aligned} \tag{7}$$

Here, the definitions of $\mathcal{N}^I(x_i)$ and $\mathcal{N}^E(x_i)$ are the same as in LDCCA. That is, we still find k nearest neighbors in original space rather than in the kernel-induced feature space. $K_x(\cdot,\cdot)$ and $K_y(\cdot,\cdot)$ are kernel functions, $(K_x)_{ij} = K_x\left(x_i, x_j\right)$ and $\left(K_y\right)_{ij} = K_y\left(y_i, y_j\right)$. The solution of Eq. 5 is similar to those in CCA and LDCCA. We omit it here due to space limit.

After getting generalized eigenvectors $(\alpha_i, \beta_i) \in \mathcal{R}^n \times \mathcal{R}^n$, we can get W_φ and W_φ as follows:

$$\begin{aligned} W_\phi &= \left[\omega_{\phi 1} \ldots \omega_{\phi d}\right] = \phi(X)\left[\alpha_1 \ldots \alpha_d\right] \\ W_\varphi &= \left[\omega_{\varphi 1} \ldots \omega_{\varphi d}\right] = \varphi(Y)\left[\beta_1 \ldots \beta_d\right] \end{aligned} \tag{8}$$

For each example (x, y), we can extract features as follows:

$$\begin{aligned} W_\phi^T \phi(x) &= \left[\alpha_1 \ldots \alpha_d\right]^T \left[K_x(x_1, x) \ldots K_x(x_n, x)\right]^T \in \mathcal{R}^d \\ W_\varphi^T \varphi(y) &= \left[\beta_1 \ldots \beta_d\right]^T \left[K_y(y_1, y) \ldots K_y(y_n, y)\right]^T \in \mathcal{R}^d \end{aligned} \tag{9}$$

With the features extracted by KLDCCA, we can then obtain the final representations through aforementioned parallel combination (PC) and serial combination (SC), respectively, (Eq. 4).

5 Experiments

In this section, we evaluate the performances of the proposed LDCCA and KLDCCA algorithms on several data sets. At first, we simply investigate the influences on classification of features extracted by LDCCA and KLDCCA on an artificial dataset. Then we discuss their classification performances on Multiple Feature database. Finally, we apply LDCCA and KLDCCA to face recognition on three face databases ORL[1], Yale[2] and AR[3]. We first perform dimensionality reduction on all data sets using LDCCA and KLDCCA as well as other related methods, CCA, PLS, LPCCA and KCCA. Then nearest neighborhood classifier is employed to estimate the classification accuracies of different methods.

In kernel-based algorithms, an important parameter to determine is type of kernel and its related parameters. We adopt the Gaussian kernel in our experiments, i.e. $K_x(x_1, x_2) = e^{-\|x_1-x_2\|^2/2\sigma_x^2}$ and $K_y(y_1, y_2) = e^{-\|y_1-y_2\|^2/2\sigma_y^2}$. Here, the kernel widths σ_x^2 and σ_y^2 are chosen by searching the following parameter space:

$$\left[2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}\right] \times \sigma_{x_o}^2 \text{ and } \left[2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}\right] \times \sigma_{y_o}^2 \tag{10}$$

where $\sigma_{x_o}^2$ and $\sigma_{y_o}^2$ denotes the mean square distances of sample X and Y, which are defined as

$$\begin{aligned}\sigma_{x_o}^2 &= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|x_i - x_j\right\|^2 \\ \sigma_{y_o}^2 &= \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\|y_i - y_j\right\|^2\end{aligned} \tag{11}$$

In order to control the relative contributions of C_ω and C_b in LDCCA and KLDCCA, a balancing factor η is introduced, which is optimized by searching in the range $\eta \in [0.001, 0.01, 0.1, 1, 10, 100]$. Besides, there are another important parameter k in LPCCA, LDCCA and KLDCCA, i.e. the number of nearest neighbors, which will be searched in the range from one to the number of training data. Furthermore, the effect of k will be demonstrated in practice in Sect. 5.4.

In our experiments, two-fold cross-validation is performed to find the optimal values of the above parameters in their respective ranges. However, it's worthy of noting that if we jointly optimize those parameters together, there will produce a huge search space, where finding the optimal values will be very time-consuming. So we optimize those parameters independently, i.e. in a 'one-by-one' manner.

For all methods, after obtaining eigenvectors ω_{xi} and ω_{yi}, we choose d eigenvectors corresponding to the d largest eigenvalues whose sum dividing the total eigenvalues is no less than a predefined threshold θ. We set $\theta = 0.95$ in all experiments.

5.1 Toy Problem

In order to intuitively review the trait on data visualization of LDCCA and KLDCCA, we consider a two-class classification problem used in [19], which contain 150 two-dimensional samples with two views $X = [X_1, X_2]$ and $Y = [Y_1, Y_2]$, where X_i and Y_i denote

[1] http://www.cl.cam.ac.uk/Research/DTG/attarchive/facedatabase.html.

[2] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[3] http://cobweb.ecn.purdue.edu/~aleix_face_DB.html.

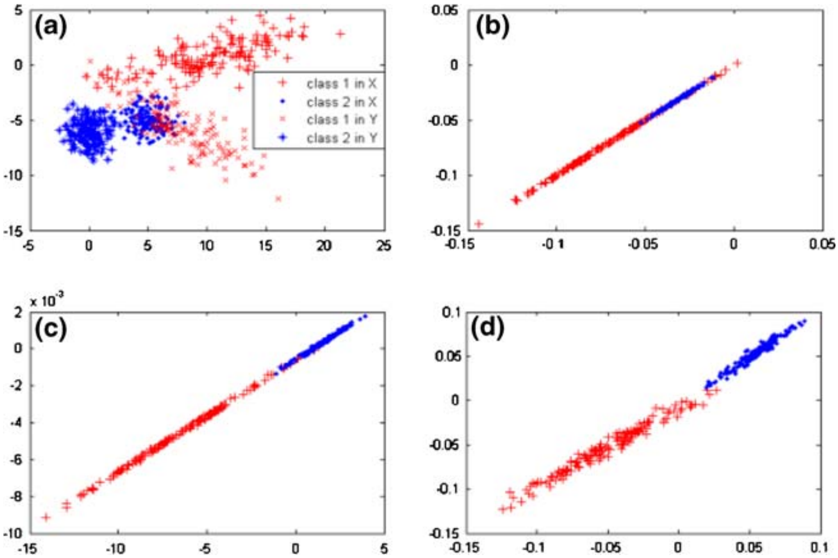

Fig. 1 Toy problems. **a** The data distribution of two classes, sign +• denote the features according to samples of the first and second class. **b, c, d** The distribution of the first pair of features extracted by CCA, LPCCA and LDCCA, respectively

ith $(i = 1, 2)$ class respectively, X_i follows the Gaussian distribution $N\left(\mu_{i,} \sum_i\right)$, where $\mu_1 = [10.18, 0.66]^T$, $\sum\limits_{1} = [15, 3.75; 3.75, 15]$, $\mu_2 = [5, -5]^T$, $\sum\limits_{2} = [1, 0; 1]$. Sample y_i is obtained from transformation as follows: $y = W^T x_i + \epsilon_i, i = 1 \ldots 150$, where $W = \left[0.6, -\sqrt{1/2}; 0.8, \sqrt{1/2}\right]$, ϵ is additional Gaussian noise whose distribution follows $N \mu_\epsilon \sum\limits_{\epsilon}$, in which $\mu_\epsilon = [1, 1]^T$, $\sum\limits_{\epsilon} = [0.01, 0; 0, 0.01]$. So, x_i and y_i satisfy linear correlation relation in some degree. Figure 1a shows the distribution of the above data. Half of the data are used for learning the projections, while the rest is for predicting.

In our experiment, we compare LDCCA and KLDCCA with CCA, KCCA and LPCCA. Figure 1b–d shows the distribution of the first feature $\left(\omega_{x1}^T X, \omega_{y1}^T Y\right)$ extracted by CCA, LPCCA and LDCCA, respectively. Figure 2a–b compare the result of KCCA with that of KLDCCA. In LPCCA, the parameter k, i.e. the number of neighbors, is set to 100, while in LDCCA and KLDCCA it is set to ten. Under those values, the above methods could achieve approximately optimal result.

From the experimental results, we can see that:

(1) CCA has revealed the linear relationships of the original features, but there is severe overlap between different classes (see Fig. 1b), which will result in poor classification performance. In our experiment, its classification accuracy is only 0.6412. In LPCCA, the overlap phenomena still exist in some degree (Fig. 1c) and the corresponding classification accuracy is 0.8680, superior to CCA;

(2) In LDCCA, two classes are separated well (Fig. 1d) and the corresponding classification accuracy can reach 0.9870. Moreover, we can see from Fig. 2 the data features extracted by KLDCCA are separated completely, and the corresponding classification accuracy is perfect 100%. In contrast, there is still some overlap in KCCA whose accuracy is

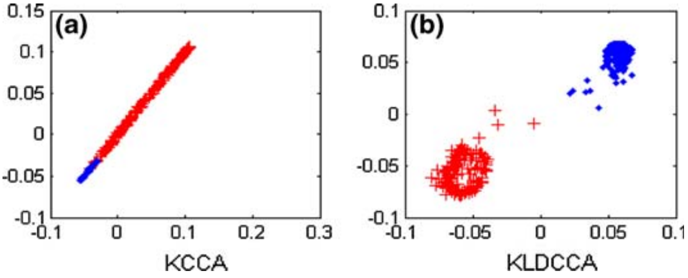

Fig. 2 Distribution of the first pair of features extracted by KCCA (**a**) and KLDCCA (**b**)

0.9512. The experimental results show that features extracted by LDCCA and KLDCCA are more beneficial for classification than others.

5.2 Multiple Feature Recognition Experiment

In this experiment, we choose Multiple Feature data set picked out from UCI machine learning repository, which consists of six sets of features of ten handwritten digits 0–9. Each class has 200 examples, so the sample size is 2,000 in total. The six sets of features extracted from multiple feature databases are flourier coefficient, contour correlation characteristics, Karhunen-Loève expansion coefficient, pixel average, Zernike moment and morphological characteristics. The name and the dimension of those features are (fou, 76), (fac, 216), (kar, 64), (pix, 240), (zer, 47) and (mor, 6).

We choose two sets of features at random as X set and Y set, so there will be fifteen $\left(C_6^2 = 15\right)$ data combination modes. For each combination, we choose 100 samples in each class for training randomly, the rest for testing. In this way, the number of training set and testing set are all 1,000. Such random experiments are repeated 10 times independently, and then we record the average recognition accuracy. We extract features by LDCCA and KLDCCA as well as CCA, LPCCA and KCCA at first. Then we perform classification based on the extracted features using nearest neighbor classifier. Table 2 gives the accuracies of those methods as well as a recently proposed method SVM-2K, which integrate KCCA and SVM in a single model [12]. In our experiments, we use the same procedure as in KLDCCA to tune kernel parameters in SVM-2K. In LPCCA, LDCCA and KLDCCA, the optimal values for parameter k are chosen. From the table, we can see that in 15 combination modes, the accuracy of LDCCA and KLDCCA are better than others in 12 combination modes. In contrast, the LPCCA provides the best result only once, and SVM-2K provides the best result twice. We also perform statistically significant tests on this database, and the results show that in most cases our methods are significantly better than the other methods.

5.3 Face Recognition Experiments

In order to verify the performances of LDCCA and KLDCCA for face recognition, we do experiments on three face databases including ORL, Yale and AR. First, we do dimensionality reduction on these face datasets, then perform classification using the nearest neighbor classifier. We compare our methods with Eigenface [23], Fisherface [24], PLS [25], CCA [26], KCCA, SVM-2K and LPCCA [18]. In CCA, KCCA, SVM-2K, LPCCA, LDCCA and KLDCCA, Daubechies orthogonal wavelet transform are repeated for two times for each image and the low frequency components are chosen as another dataset [26]. Before

Table 2 Classification accuracy on multiple feature database

		KLDCCA		LDCCA		LPCCA		SVM-2K		KCCA		CCA	
X	*Y*	PR1	PR2	PR1 (+ \)	PR2 (0 \)	PR1 (+ +)	PR2 (+ +)	PR1 (+ 0)	PR2 (0 +)	PR1 (+ +)	PR2 (+ +)	PR1 (+ +)	PR2 (+ +)
fac	fou	**0.9720**	0.9650	0.9629	**0.9830**	0.9614	0.9700	0.9540	0.9570	0.9174	0.9268	0.8792	0.8899
fac	kar	0.9790	0.9690	**0.9803**	**0.9827**	0.9690	0.9663	0.9710	0.9789	0.9639	0.9587	0.9648	0.9649
fac	mor	**0.9780**	**0.9800**	0.9074	0.9289	0.7993	0.8422	0.9117	0.9212	0.8950	0.9056	0.7605	0.7691
fac	pix	0.9760	0.9390	**0.9760**	0.9791	0.9657	0.9679	0.9751	**0.9857**	0.9651	0.9657	0.9524	0.9514
fac	zer	0.9760	0.9730	0.9679	**0.9800**	0.9606	0.9596	**0.9850**	0.9770	0.9615	0.9542	0.8597	0.8698
fou	kar	**0.9740**	0.9570	0.9578	**0.9763**	0.9152	0.9383	0.9466	0.9436	0.9160	0.9160	0.9069	0.9294
fou	mor	**0.8600**	**0.8630**	0.8124	0.8292	0.7637	0.7696	0.7790	0.7899	0.6703	0.6741	0.7602	0.7693
fou	pix	**0.9600**	0.8670	0.9469	**0.9694**	0.8484	0.8641	0.9377	0.9472	0.9100	0.9220	0.8377	0.8519
fou	zer	0.8600	0.8360	0.8525	0.8636	0.8387	0.8482	**0.8786**	**0.8779**	0.8368	0.8397	0.8332	0.8424
kar	mor	0.9760	**0.9830**	0.8923	0.9262	0.8004	0.8460	0.8490	0.8900	0.7380	0.7800	0.7871	0.8174
kar	pix	0.9630	0.9360	0.9654	0.9648	**0.9701**	**0.9703**	0.9404	0.9508	0.8906	0.8906	0.9675	0.9673
kar	zer	**0.9770**	0.9670	0.9571	**0.9701**	0.9569	0.9687	0.9540	0.9533	0.9120	0.9030	0.9144	0.9263
mor	pix	**0.9710**	**0.9760**	0.8773	0.9114	0.7439	0.7740	0.8786	0.8900	0.8160	0.8560	0.7295	0.7601
mor	zer	**0.8320**	**0.8400**	0.7972	0.8135	0.7341	0.7561	0.8150	0.8055	0.7605	0.7454	0.7410	0.7580
pix	zer	**0.9670**	0.9490	0.9445	0.9629	0.9320	0.9426	0.9395	0.9477	0.9080	0.9240	0.8331	0.8506
AVE		**0.9481**	0.9333	0.9199	**0.9361**	0.8773	0.8923	0.9143	0.9210	0.8707	0.8775	0.8485	0.8612

Note: PR1 and PR2 denote respectively, accuracies of features by parallel combination and serial combination; the first symbol in the bracket denotes significance results across 15 combinations (*t*-tests at 95% significance level) between KLDCCA and others, while the second one denotes significance results between LDCCA and others; + significantly better; 0 neither significantly better nor worse; \ not applicable doing this, PCA is applied to the datasets to reduce dimensionality to avoid small sample problem [26].

5.3.1 ORL Database

ORL dataset, also called AT & T face dataset, contains images of 40 person, and everyone has ten images which are photographed in different time and illumination with different facial expression (eye opening or not, smile or not) and facial detail (wear glasses or not). The images are all sized 112 × 92 pixels with a 256-level gray scale, and the background is uniform black.

For each person, five images are chosen randomly for training and the rest images for classification. So there are 200 training images and 200 testing images in total. The experiment is repeated ten times and the average accuracy is recorded. Table 3 lists the accuracies of different methods on ORL dataset. In PLS, CCA, KCCA, LPCCA, LDCCA and KLDCCA, the parallel combination (PC) and the serial combination (SC) mode are all evaluated. In LPCCA, LDCCA and KLDCCA, the optimal number of neighbor k is also listed in the table. The table shows that LDCCA is comparable to PLS, KCCA and SVM-2K, and is significantly better and CCA and LPCCA. On this database, KLDCCA achieves the highest accuracy.

5.3.2 Yale Database

Yale dataset contains 165 gray level images of 15 person, 11 images for each person, including left/right/front illumination, wearing glasses or not, normal face, happiness, sadness, sleepiness, surprise, blink and so on.

We crop the original image of Yale dataset into 100 × 100 size through manual calibrate mode. For each individual, we choose six images for training and the rest images for classification. So there are 90 training samples and 75 testing samples in total. Similarly, the experiment is repeated ten times and the average result is recorded finally. Table 4 gives the accuracies of different methods on this database. In LPCCA, LDCCA and KLDCCA, the optimal number of neighbor k are also listed in the table. From Table 4, we can see that LDCCA achieves better result over other methods significantly. It's also noteworthy that both KCCA and KLDCCA are inferior to corresponding CCA and LDCCA on this database.

Table 3 Classification accuracy on ORL database

Methods	Classification accuracy			
Eigenface	0.9312			
Fisherface	0.9005			
/	PR1	k	PR2	k
PLS	0.9401	/	0.9398	/
CCA	0.9031	/	0.9011	/
KCCA	0.9485	/	0.9485	/
SVM-2K	0.967	/	0.9511	/
LPCCA	0.9035	198	0.9035	198
LDCCA	0.9475	4	0.9489	7
KLDCCA	**0.9725**	4	**0.9555**	4

Table 4 Classification accuracy on Yale database

Methods	Classification accuracy			
Eigenface	0.6925			
Fisherface	0.9133			
/	PR1	*k*	PR2	*k*
PLS	0.7174	/	0.7174	/
CCA	0.8533	/	0.8533	/
KCCA	0.7483	/	0.7521	/
SVM_2K	0.8851	/	0.8851	/
LPCCA	0.7613	4	0.7653	3
LDCCA	**0.9560**	2	**0.9560**	2
KLDCCA	0.9333	4	0.7868	4

Table 5 Classification accuracy on AR database

Methods	Classification accuracy			
Eigenface	0.7011			
Fisherface	0.9644			
/	PR1	*k*	PR2	*k*
PLS	0.6671	/	0.6721	/
CCA	0.9100	/	0.9100	/
KCCA	0.8735	/	0.8854	/
SVM_2K	0.9412	/	0.9412	/
LPCCA	0.8463	342	0.8451	342
LDCCA	0.9746	6	**0.9763**	7
KLDCCA	**0.9869**	6	0.8889	7

We guess one possible reason may be that the size of the database is too small and thus those kernel methods are prone to over-fitting due to the difficulty of kernel parameters tuning.

5.3.3 AR Database

AR dataset contains nearly 4,000 color images of 126 people (70 men and 56 women). Each person has 26 images belonging to two groups, each of which has 13 images and the photographic time has 2 weeks interval. The two groups show, respectively, the variety of facial expression, illumination or occlusion (sunglasses or shawl).

We randomly choose 50 people (30 men and 20 women), and do experiment with each person's 14 images which are no occlusion, thus, we have a face database with 700 images. After clipping, scale tension, manual calibrating and so on, we dispose each image into 165 × 120 pixels. We randomly choose seven images for training and the other for identification, so we have 350 training samples and 350 testing samples each time. The random experiments are performed ten times and the average accuracy is shown in Table 5.

Table 5 gives the accuracies of different algorithms on AR database. In LPCCA, LDCCA and KLDCCA, the optimal values of neighbor *k* are also listed in the table. From the table,

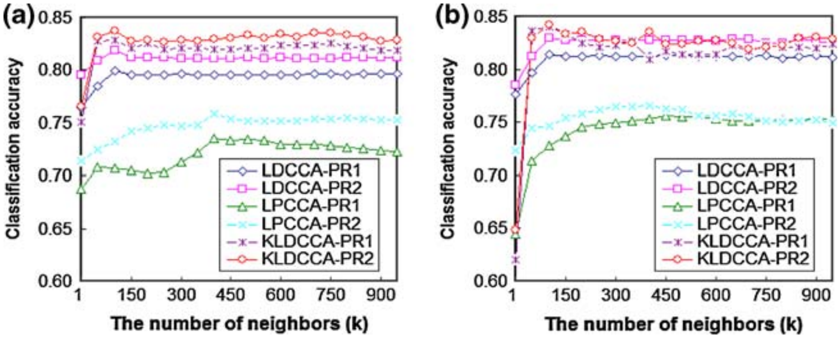

Fig. 3 Classification accuracy under the variation of k on the combination of fou and mor (**a**) and on the combination of mor and zer (**b**)

we can see that both LDCCA and KLDCCA achieve competitive results over other methods. Moreover, we notice that the optimal numbers of neighbors in LDCCA and KLDCCA are much smaller than that in LPCCA. Finally as on the Yale database, KLDCCA under serial combination and KCCA under both parallel and serial combinations show inferior performances than LDCCA and CCA, respectively.

5.4 Effect of Parameter k

One important parameter in LDCCA and KLDCCA, as well as LPCCA is the number of neighbors used by k-NN. However, to the best of our knowledge, it's challenging to determine the appropriate value of k in advance, so do in most locality based methods [15–17]. In our experiments, we determine the optimal value of the parameter by searching in the range from one to the number of training samples. Figure 3 shows the effect of the number of neighbors on those methods, LPCCA, LDCCA and KLDCCA. At first, features are extracted using LPCCA, LDCCA and KLDCCA with different values of the parameter k. Then we perform classification based on the extracted features using nearest neighbor classifier. Figure 3 gives the corresponding classification accuracy under the combination of fou and mor, and the combination of mor and zer, respectively. As before, the results of the two feature fusion strategies are all included. From Fig. 3, we can see that the curves corresponding to LDCCA and KLDCCA are much smoother than those of LPCCA. It implies that the performances of LDCCA and KLDCCA are less sensitive to the selection of the parameter k than that of LPCCA. Furthermore, it can be found from Fig. 3 and Tables 3–5 that the optimal values of k in LDCCA and KLDCCA are much smaller than that in LPCCA in most cases. The reason for that interesting phenomenon may be that LDCCA and KLDCCA do not find their neighbors in the entire training set, but confine the search in specific classes. On the other hand, LDCCA and KLDCCA aim at keeping the local discriminative information of data, while LPCCA aims at keeping the local manifold structure of data. We guess that the latter may need more neighbors than the former to achieve a satisfying performance. Our experiments suggest that when choosing the optimal values of k for LDCCA and KLDCCA in practice, we do not have to search the entire range from one to the number of training samples but only need to investigate a small subset with smaller values. Thus, the computational cost can be greatly reduced.

6 Conclusion

In this paper, we propose a new CCA model with local discrimination, called LDCCA. Different from CCA, LDCCA considers not only the correlations between the sample pairs but also the correlations between samples and their local neighborhoods. We design a new objective function in LDCCA by maximizing local within-class correlations and minimizing local between-class correlations simultaneously. What is more, a kernel generalization of LDCCA is also developed to cope with nonlinear problems. Extensive experiments on a series of data sets show that the proposed methods including LDCCA and KLDCCA can effectively improve the classification performance.

In KLDCCA, we find k nearest neighbors in original space rather than in the kernel-induced feature space. It's interesting to investigate whether we can further improve the performances if we identify neighbors based on the kernel-induced distance metric. Besides, in the current study the parameters in our methods are determined in a "one-by-one" manner, and it would be better to jointly optimize them. So it is worth exploring effective and efficient optimizing approaches in future work. Also, it's interesting to further explore why LDCCA needs smaller number of nearest neighbors than LPCCA. Finally, we plan to combine the ideas of LDCCA and KLDCCA with multi-set CCA [27] for multi-modal problem in our future work.

Acknowledgments The authors would like to thank the anonymous referees for their helpful comments and suggestions to improve the presentation of this paper. This work is partly supported by National Science Foundation of China under Grant Nos. 60875030 and 60603029.

Appendix : The Solution of LDCCA

The solution of LDCCA is equivalent to the optimal problem as follows:

$$\begin{aligned} &\max_{\omega_x,\omega_y} \omega_x^T \tilde{C}_{xy}\omega_y \\ &s.t. \omega_x^T C_{xx}\omega_x = 1, \omega_y^T C_{YY}\omega_Y = 1 \end{aligned} \tag{A-1}$$

Similarly as in CCA [9], the LDCCA equation can be rewritten as:

$$\begin{aligned} \tilde{C}_{xy}C_{yy}^{-1}\tilde{C}_{yx}\omega_x &= \lambda^2 C_{xx}\omega_x \\ \tilde{C}_{yx}C_{xx}^{-1}\tilde{C}_{xy}\omega_y &= \lambda^2 C_{yy}\omega_y \end{aligned} \tag{A-2}$$

In this paper, we use singular value decomposition (SVD) to solve LDCCA equation following [28]. Let $H = C_{xx}^{-1/2}\tilde{C}_{xy}C_{yy}^{-1/2}$, $u = C_{xx}^{1/2}\omega_x$, $v = C_{yy}^{1/2}\omega y$, then Eq. A-2 can be rewritten as:

$$\begin{cases} HH^T u = \lambda^2 u \\ H^T H v = \lambda^2 v \end{cases} \tag{A-3}$$

Let $H = UDV^T = \sum_{i=1}^{d} u_i v_i^T$ be the SVD decomposition of matrix H, where the i-th diagonal element of diagonal matrix D is just λ_i, u_i and v_i are respectively the i th row of matrix U and V, corresponding to singular value λ_i, we have

$$\begin{cases} \omega_{xi} = C_{xx}^{-1/2} u_i \\ \omega_{yi} = C_{yy}^{-1/2} v_i \end{cases} \tag{A-4}$$

From Eq. A-4, we obtain the ith ($i = 1 \ldots d$) pairs of basis vector of LDCCA.

References

1. Hotelling H (1936) Relation between two sets of variates. Biometrika 28(3):321–377
2. Gao H, Hong W, Cui J, Xu Y (2007) Optimization of principal component analysis in feature extraction. In: IEEE international conference mechatronics and automation, pp 3128–3132
3. Zheng W, Zhou X, Zou C et al. (2006) Facial expression recognition using kernel canonical correlation analysis. IEEE Trans Neural Netw 17(1):233–238
4. Nielsen AA (2002) Multiset canonical correlations analysis and multispectral truly multitemporal remote sensing data, image processing. IEEE Trans Image Process 11(3):293–305
5. Vlassis N, Motomura Y, Krosa B (2000) Supervised linear feature extraction for mobile robot localization. In: Proceedings of the IEEE international conference on robotics and automation, pp 2979–2984
6. Melzer T, Reiter M, Bischof H (2003) Appearance models based on kernel canonical correlation analysis. Pattern Recognit 36(9):1961–1971
7. Abraham B, Merola G (2005) Dimensionality reduction approach to multivariate prediction. Comput Stat Data Anal 48(1):5–16
8. Horikawa Y (2004) Use of autocorrelation kernels in kernel canonical correlation analysis for text classification. In: International conference on neural information processing, pp 1235–1240
9. Hardoon DR, Szedmak S, Shawe-Taylor J (2004) Canconical correlation analysis: an overview with application to learning methods. Neural Comput 16(12):2639–2664
10. Li Y, Shawe-Taylor J (2006) Using KCCA for Japanese–English cross-language information retrieval and document classification. J Intell Inf Syst 27(2):117–133
11. Fukumizu K, Bach FR, Gretton A (2007) Statistical consistency of kernel canonical correlation analysis. J Mach Learn Res 8:361–383
12. Farquhar JDR, Hardoon DR, Meng H, Shawe-Taylor J, Szedmak S (2005) Two view learning: SVM-2K, theory and practice. Adv Neural Inf Process Syst
13. Hsieh W (2000) Nonlinear canonical correlation analysis by neural network. Neural Netw 13(10):1095–1105
14. Kambhatla N, Leen TK (1997) Dimension reduction by local principal component analysis. Neural Comput 9:1493–1516
15. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290:2323–2326
16. Tenenbaum J, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290:2319–2323
17. He X, Niyogi P (2003) Locality preserving projections. Adv Neural Inf Process Syst 16, Vancouver, Canada
18. Sun T, Chen S (2007) Locality preserving CCA with applications to data visualization and pose estimation. Image and Vis Comput 25(5):531–543
19. Sun T (2006) Enhanced canonical correlation analysis and application. PhD dissertation, Nanjing University of Aeronautics and Astronautics
20. Shawe-Taylor J, Williams CKI, Cristianini N, Kandola JS (2005) On the eigenspectrum of the gram matrix and the generalization error of kernel-PCA. IEEE Trans Inf Theory 51(7): 2510–2522
21. Bach FR, Jordan MI (2002) Kernel independent component analysis. J Mach Learn Res 3:1–48
22. Bach FR, Jordan MI (2002) Learning graphical models with mercer kernels. In: Neural information processing systems, vol 15, pp 1009–1016
23. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):17–166173
24. Belhumeour 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
25. Sun Q, Jin Z, Heng PA, et al. (2005) A novel feature fusion method based on partial least squares regression. In: International conference on advances in pattern recognition, pp 268–277
26. Sun Q, Zeng S, Liu Y, Heng PA, Xia DS (2005) A new method of feature fusion and its application in image recognition. Pattern Recognit 38(12):2437–2448

27. Via J, Santamaria I, Perez J (2007) A learning algorithm for adaptive canonical correlation analysis of several data sets. Neural Netw 20(1):139–152
28. Sun T, Chen S, Yang JY, Shi P (2008) A novel method of combined feature extraction for recognition. In: IEEE conferences on data mining, pp 1043–1048