Multi-view Uncorrelated Discriminant Analysis

Shiliang Sun, Xijiong Xie, and Mo Yang

Abstract—Multi-view learning is more robust than single-view learning in many real applications. Canonical Correlation Analysis (CCA) is a popular technique to utilize information stemming from multiple feature sets. However, it does not exploit label information effectively. Later Multi-view Linear Discriminant Analysis (MLDA) was proposed through combining CCA and Linear Discriminant Analysis (LDA). Due to the successful application of Uncorrelated Linear Discriminant Analysis (ULDA), which seeks optimal discriminant features with minimum redundancy, we propose a new supervised learning method called Multi-view Uncorrelated Linear Discriminant Analysis (MULDA) in this paper. This method combines the theory of ULDA with CCA. Then we adapt Discriminant Canonical Correlation Analysis (DCCA) instead of the CCA in MLDA and MULDA, and discuss about the effect of this modification. Furthermore, we generalize these methods to the nonlinear case by kernel-based learning techniques. The new method is called Kernel Multi-view Uncorrelated Discriminant Analysis (KMUDA). Then we modify Kernel Multi-view Discriminant Analysis (KMDA) and KMUDA by replacing Kernel Canonical Correlation Analysis (KCCA) with Kernel Discriminant Canonical Correlation Analysis (KDCCA). Our methods are tested on different real datasets and compared with other state-of-the-art methods. Experimental results validate the effectiveness of our methods.

Index Terms—Feature extraction, Multi-view discriminant analysis, Uncorrelated discriminant analysis, Canonical correlation analysis, Kernel-based learning technique.

1 Introduction

In the real world, an object can be observed from different viewpoints, which indicates that it can be described by multiple distinct feature sets. However, learning from a single view may be non-robust. Motivated by these reasons, multi-view learning [1] was proposed. A critical issue in multi-view learning is to effectively utilize the information stemming from different feature sets. One effective approach is to fuse information through obtaining a common subspace for these feature sets and feature extraction is often used to achieve this subspace.

Canonical Correlation Analysis (CCA), first proposed by Hotelling [2], is a powerful tool for feature extraction in multi-view learning. It works on paired datasets to find two linear transformations each for one view such that the two transformed variables are most correlated. However, an inherent shortage of CCA is that label information is not utilized, which may limit it in the classification performance. Linear Discriminant Analysis (LDA) [3][4] is a popular supervised learning method in single-view learning. It seeks an optimal linear transformation that maps data into a subspace, in which the within-class distance is minimized and the between-class distance is maximized simultaneously. Following the way LDA preserves the class structure, Discriminant CCA (DCCA) [5] was proposed to exploit the discriminant structure in multi-view learning. It takes within-class and between-class correlation terms from different views into account, and therefore the inter-view class structure can be preserved. Another approach to utilizing label information in multi-view learning is realized by maximizing the consistency between predicted labels. For example, Multi-view Fisher Discriminant Analysis (MFDA) [6][7] was proposed to learn classifiers in different views. The difference between the predicted labels of these classifiers is minimized. However, it can only be applied in binary classification. Later Chen and Sun [8] used a hierarchical clustering approach to extend MFDA to the multi-class scenario, namely Hierarchical MFDA (HMFDA).

As mentioned above, preserving the discriminant structure is very important in feature extraction. In the scenario of multi-view learning, both inter-view and intra-view discriminant information are important to ensure the classification performance in the common subspace. DCCA, as we introduced in the last paragraph, just takes cross-view correlation into account, which means the inter-view class structure is preserved, while the intra-view data structure is ignored yet. Multi-view Discriminant Analysis (MvDA) [9] is an effective method to cope with this issue. It maximizes the difference between the within-class variation and the between-class variation which are calculated from the examples across all views. It can be cast as a natural extension of LDA with all the transformed feature sets (e.g. different views) regarded as a large dataset.

Multi-view Linear Discriminant Analysis (MLDA) [10][23] can be regarded as a combination of CCA and LDA. Through optimizing the corresponding objective, discrimination in each view and correlation between two views can be maximized simultaneously. Uncorrelated LDA (ULDA) [11][12][13][14] is an extension of LDA by adding some constraints into the optimization objective of LDA, so that the feature vectors extracted by ULDA could contain minimum redundancy. Similarly, motivated by the successful application of ULDA in various applications, we propose Multi-view Uncorrelated Linear Discriminant Analysis (MULDA) by imposing two

• *Shiliang Sun (corresponding author), Xijiong Xie and Mo Yang are with Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, P. R. China (email: shiliangsun@gmail.com, slsun@cs.ecnu.edu.cn)*

Manuscript received 1 Dec. 2014, revised 13 Mar. 2015, revised 29 Apr. 2015 and 13 Sep. 2015, accepted 17 Nov. 2015.

more constraints in each view. It extracts uncorrelated features in each view and computes transformations of each view to project data into a common subspace.

Part of this research has been reported in a short conference paper [10]. Except the above work, there are mainly two differences in this paper compared to the previous work. Firstly since DCCA is able to preserve class structures between two views and the corresponding objective is similar to CCA, the CCA part is further replaced with DCCA in MLDA and MULDA. The effect of this modification is shown in the experimental results. Secondly as all the methods mentioned before are linear methods, when data have weak linear separability, the performance of these methods may be poor. Kernel-based learning techniques are a feasible approach to deal with the nonlinear problem. They map the input space into a high dimensional feature space, in which a nonlinear problem can be solved as a traditional linear problem. Even though the problem can be solved by linear methods, kernel extensions of linear methods can often provide better performance. For example, Kernel CCA (KCCA) [15][16][17] was provided as a nonlinear extension of CCA by means of the kernel trick. In [18], Generalized Discriminant Analysis (GDA) was proposed to generalize linear discriminant analysis to kernel-based nonlinear discriminant analysis and MLDA was also extended to Kernel Multi-view Discriminant Analysis (KMDA) [23]. Uncorrelated discriminant vectors using the kernel method were proposed in [19] to extend ULDA. Similarly, DCCA has its nonlinear version called Kernelized Discriminative Canonical Correlation Analysis (KDCCA) [20]. Thus we propose a new method called Kernel Multi-view Uncorrelated Discriminant Analysis (KMUDA). It is expressed by the kernel operators and can similarly be regarded as a combination of GDA and KCCA. As we tried before, we will also replace the KCCA part with KDCCA in KMDA and KMUDA, and study the effect.

In the next section, we review some related work briefly. Then the formulations and solutions of MULDA are presented in Section 3. Furthermore, the modifications of MLDA and MULDA are also presented in this section. Finally, we provide the time complexity of the linear feature extraction algorithms. Section 4 gives the explicit objective of KMUDA and the derivation of the corresponding closed-form solution, where the modifications of KMDA and KMUDA are also given. Then we provide the time complexity of the nonlinear feature extraction algorithms. After reporting experimental results in Section 5, we conclude this paper and discuss some future work in Section 6.

2 Related Work

In this section, first some basic notations that will be used are presented. Then we give a brief review of some research related to our work.

2.1 Notations

Let X and Y be two normalized feature matrices whose mean values are 0, respectively. $X = [x_1, x_2, ..., x_n] = [X_1, X_2, ..., X_k], X \in \mathbb{R}^{p\times n}$, where $x_j \in \mathbb{R}^p\,(1 \le j \le n)$ represents an example, n is the number of examples, m is the number of classes and $X_i \in \mathbb{R}^{p\times n_i}$ denotes the subset of all the examples in class i with n_i being the number of examples in this subset. Similarly, $Y = [y_1, y_2, ..., y_n] = [Y_1, Y_2, ..., Y_k], Y \in \mathbb{R}^{q\times n}$. Then we have a two-view dataset $\{(x_1, y_1), ..., (x_n, y_n)\}$. In the remainder of this paper, when we refer to single-view learning, view X is used.

For kernel methods, we need to map the feature sets into a Hilbert space F. Suppose we have two nonlinear mapping functions $\phi_x : \mathbb{R}^p \to F,\ x_j \mapsto \phi_x(x_j)$ and $\phi_y : \mathbb{R}^q \to F,\ y_j \mapsto \phi_y(y_j)$. Then X and Y are mapped into $\phi_x(X) = [\phi_x(x_1), ..., \phi_x(x_n)]$ and $\phi_y(Y) = [\phi_y(y_1), ..., \phi_y(y_n)]$, respectively. We assume that the examples in X and Y are centered in F for convenience (the mapped examples can be mean-normalized by using the method in [18]). In order to generalize linear methods to the nonlinear case, the inner product is replaced with the following Mercer kernel function: $k(x_i, x_j) = \phi_x(x_i)^T\phi_x(x_j)$. So kernel matrices can be represented as $K_x = \phi_x(X)^T\phi_x(X)$ and $K_y = \phi_y(Y)^T\phi_y(Y)$.

2.2 CCA and KCCA

CCA is an approach to correlating linear relationships between two-view feature sets [17]. It seeks linear transformations each for one view such that the correlation between these transformed feature sets are maximized in the common subspace.

The aim of CCA is to find two projection directions w_x and w_y, one for each view, and the following linear correlation coefficient

$$\frac{cov\left(w_x^TX, w_y^TY\right)}{\sqrt{var\left(w_x^TX\right)var\left(w_y^TY\right)}} = \frac{w_x^TC_{xy}w_y}{\sqrt{\left(w_x^TC_{xx}w_x\right)\left(w_y^TC_{yy}w_y\right)}} \tag{1}$$

is maximized. In this equation (1), the covariance matrices C_{xy}, C_{xx} and C_{yy} are calculated as

$$C_{xy} = \frac{1}{n}XY^T,\ C_{xx} = \frac{1}{n}XX^T,\ C_{yy} = \frac{1}{n}YY^T. \tag{2}$$

The term $\frac{1}{n}$ in (2) can be cancelled out when calculating the correlation coefficient. We omit it from these expressions in the remainder of this paper. Since w_x, w_y are scale-independent, (1) is equivalent to the following optimization problem

$$\begin{array}{ll}\max_{w_x,w_y} & w_x^TC_{xy}w_y \\ \text{s.t.} & w_x^TC_{xx}w_x = 1,\ w_y^TC_{yy}w_y = 1.\end{array} \tag{3}$$

It can be transformed into a generalized eigenvalue problem as

$$\begin{bmatrix}\mathbf{0} & C_{xy}\\ C_{yx} & \mathbf{0}\end{bmatrix}\begin{bmatrix}w_x\\ w_y\end{bmatrix} = \lambda\begin{bmatrix}C_{xx} & \mathbf{0}\\ \mathbf{0} & C_{yy}\end{bmatrix}\begin{bmatrix}w_x\\ w_y\end{bmatrix}. \tag{4}$$

In this paper, $\mathbf{0}$ represents the appropriate number of zero elements.

KCCA [15][16][17] is a nonlinear extension of CCA. The desired projection vectors w_x^ϕ and w_y^ϕ can be expressed as a linear combination of all training examples in the feature space, and there exist coefficient vectors $a = [a^1, ..., a^n]^\top$ and $b = [b^1, ..., b^n]^\top$, such that

$$w_x^\phi = \sum_{i=1}^n a^i\phi_x(x_i) = \phi(X)a,\ w_y^\phi = \sum_{i=1}^n b^i\phi_y(y_i) = \phi(Y)b. \tag{5}$$

Substituting (5) and (2) into (3) and using the definition of the kernel matrix, one can formulate the optimization problem of KCCA as

$$\begin{array}{ll} \max_{a,b} & a^T K_x K_y b \\ \text{s.t.} & a^T K_x K_x a = 1,\ b^T K_y K_y b = 1, \end{array} \tag{6}$$

which can be solved in a similar way like CCA [2].

2.3 LDA, ULDA and GDA

LDA is an effective supervised feature extraction method for single-view learning. It seeks an optimal linear transformation to map the data into a subspace so that the ratio between between-class distance and within-class distance is maximized. The optimal transformation can be obtained by maximizing the Fisher criterion function. Given a data matrix X, the Fisher criterion function is defined as

$$F(w) = \frac{w^T S_b w}{w^T S_w w}, \tag{7}$$

where w represents the projection vector. An alternative criterion for classical LDA is

$$F(w) = \frac{w^T S_b w}{w^T S_t w}, \tag{8}$$

where S_b, S_w and S_t denote the between-class, within-class and total scatter matrix, respectively. These scatter matrices are calculated as

$$S_w = \frac{1}{n} X(I - W)X^T,\ S_b = \frac{1}{n} XWX^T,\ S_t = \frac{1}{n} XX^T, \tag{9}$$

where $W = diag(W_1, W_2, ..., W_k)$, and W_i is an $(n_i \times n_i)$ matrix with all elements equal to $\frac{1}{n_i}$. The term $\frac{1}{n}$ in these expressions is also omitted in our following work.

Similar to CCA, the optimization problem of criterion (8) can be transformed to

$$\begin{array}{ll} \max_{w} & w^T S_b w \\ \text{s.t.} & w^T S_t w = 1. \end{array} \tag{10}$$

The optimal vector w is the eigenvector corresponding to the maximum eigenvalue of $S_t^{-1} S_b$.

ULDA was first proposed in [11] to find the optimal projection vectors that are S_t-orthogonal. Specifically, to extend LDA to ULDA, we just need to add some constraints $(w_r^T S_t w_i = 0, i = 1, ..., r-1)$ into (10), so that the feature vectors extracted by ULDA can be mutually uncorrelated.

In [11], w_i is found successively as follows. The j^{th} discriminant vector w_j of ULDA is the eigenvector corresponding to the maximum eigenvalue of the following generalized eigenvalue problem

$$P_j S_b w_j = \lambda_j S_w w_j, \tag{11}$$

where

$$\begin{array}{l} P_1 = I_p, \\ P_j = I_p - S_t D_j^T (D_j S_t S_w^{-1} S_t D_j^T)^{-1} D_j S_t S_w^{-1} (j > 1), \\ D_j = [w_1, w_2, ..., w_{j-1}]^T\ (j > 1), \\ I_p = diag(1, 1, ..., 1) \in \mathbb{R}^{p \times p}. \end{array} \tag{12}$$

Based on the kernel technique ($K = \phi_x(X)^T \phi_x(X)$), the Fisher criterion (8) can be generalized to the kernel-based version GDA [18] as

$$F(w) = \frac{w^T S_b^\phi w}{w^T S_t^\phi w} = \frac{a^T KWKa}{a^T KKa}. \tag{13}$$

2.4 DCCA and KDCCA

DCCA proposed in [5] exploits class structures by taking both within-class and between-class correlation into consideration. It can preserve class structures between two views. The optimization problem of DCCA is formulated as

$$\begin{array}{ll} \max_{w_x, w_y} & w_x^T X A Y^T w_y \\ \text{s.t.} & w_x^T X X^T w_x = 1,\ w_y^T Y Y^T w_y = 1, \end{array} \tag{14}$$

where

$$A = \begin{bmatrix} 1_{n_1 \times n_1} & & & & \\ & \ddots & & & \mathbf{0} \\ & & 1_{n_i \times n_i} & & \\ & \mathbf{0} & & \ddots & \\ & & & & 1_{n_k \times n_k} \end{bmatrix}. \tag{15}$$

Applying the Lagrangian multiplier method, the solution of (14) can be transformed into a generalized eigenvalue problem

$$\begin{bmatrix} \mathbf{0} & XAY^T \\ YAX^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix} = \lambda \begin{bmatrix} XX^T & \mathbf{0} \\ \mathbf{0} & YY^T \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix}. \tag{16}$$

KDCCA [20] integrates the kernel trick into DCCA, for which the optimization problem is formulated as

$$\begin{array}{ll} \max_{a,b} & a^T K_x A K_y b \\ \text{s.t.} & a^T K_x K_x a = 1,\ b^T K_y K_y b = 1. \end{array} \tag{17}$$

3 Multi-view Uncorrelated Linear Discriminant Analysis

Inspired by the effectiveness of CCA and LDA, MLDA was proposed to incorporate these two methods. The correlation information between views and discriminant information in each view can be preserved simultaneously in the transformed common subspace. Furthermore, since ULDA can extract uncorrelated features with minimum redundancy, which may be highly desirable in many applications, we extend MLDA to a new method called MULDA. The purpose of this method is to take advantage of both CCA and ULDA, so that useful features can be exploited for multi-view applications. As we introduced in Section 2, DCCA can preserve discriminant structures between views. In this paper, we further replace the CCA part with DCCA in MLDA and MULDA, and thus both intra-view and inter-view class structures can be preserved.

In this section, first the optimization objective and corresponding solution of MLDA are introduced in Section 3.1. Then we provide the optimization problem of MULDA and state several related theorems in Section 3.2. In Section 3.3, we provide modifications of MLDA and MULDA, so that discriminant information can be preserved between views. In Section 3.4, we provide the time complexity analysis of the linear feature extraction algorithms.

3.1 Multi-view Linear Discriminant Analysis

From (2) and (9), C_{xx} and S_t both represent the total scatter matrix. MLDA was proposed to incorporate (3) and (10). The optimization problem of MLDA is given by

$$\begin{array}{ll}\max_{w_x,w_y} & w_x^T S_{b_x} w_x + w_y^T S_{b_y} w_y + 2\gamma w_x^T C_{xy} w_y \\ \text{s.t.} & w_x^T S_{t_x} w_x = 1,\ w_y^T S_{t_y} w_y = 1,\end{array} \tag{18}$$

where the matrices S_{b_x}, S_{b_y}, S_{t_x} and S_{t_y} are constructed according to (9), and C_{xy} is computed following (2).

Through optimizing (18), the correlation between different views and the discrimination of each view can be maximized simultaneously. Using the Lagrangian multiplier technique, (18) can be solved by a generalized multivariate eigenvalue problem in the following form

$$\begin{bmatrix} S_{b_x} & \gamma C_{xy} \\ \gamma C_{yx} & S_{b_y} \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix} = \begin{bmatrix} S_{t_x} & \mathbf{0} \\ \mathbf{0} & S_{t_y} \end{bmatrix} \begin{bmatrix} \lambda_x w_x \\ \lambda_y w_y \end{bmatrix}, \tag{19}$$

which has appeared in the solution of [21] and can be solved by an alternation method [22].

In order to obtain a closed-form solution, the constraints in (18) can be coupled with $\sigma = \frac{tr(S_{t_x})}{tr(S_{t_y})}$, such that the constraints are transformed into a single constraint $w_x^T S_{t_x} w_x + \sigma w_y^T S_{t_y} w_y = 1$. In the remainder, we will use this coupled constraint in our optimization problem.

3.2 Multi-view Uncorrelated Linear Discriminant Analysis

It has been proved that uncorrelated features with minimum redundancy are desirable in many applications [11][12][13][14]. Motivated by the fact that ULDA can be combined with other methods to enhance performance [24], a new approach MULDA is proposed. The extracted feature vectors will be mutually uncorrelated in each view.

Let (w_{x1}, w_{y1}) represent the vector pair solved by MLDA corresponding to the maximum eigenvalue. Suppose $r-1$ vector pairs (w_{xj}, w_{yj}), $j = 1, 2, ..., r-1$ of the two-view dataset are obtained. MULDA aims to find the r^{th} discriminant vector pair (w_{xr}, w_{yr}) of matrices X and Y which optimizes the objective function (18) and subject to the following conjugate orthogonality constraints

$$w_{xr}^T S_{t_x} w_{xj} = w_{yr}^T S_{t_y} w_{yj} = 0 \quad (j = 1, 2, ..., r-1). \tag{20}$$

The optimization problem of MULDA can be formulated as

$$\begin{array}{ll}\max_{w_{xr},w_{yr}} & w_{xr}^T S_{b_x} w_{xr} + w_{yr}^T S_{b_y} w_{yr} + 2\gamma w_{xr}^T C_{xy} w_{yr} \\ \text{s.t.} & w_{xr}^T S_{t_x} w_{xr} + \sigma w_{yr}^T S_{t_y} w_{yr} = 1, \\ & w_{xr}^T S_{t_x} w_{xj} = w_{yr}^T S_{t_y} w_{yj} = 0 \\ & (j = 1, 2, ..., r-1),\end{array} \tag{21}$$

where w_{xr} and w_{yr} represent the r^{th} discriminant vectors of matrices X and Y, respectively.

Through optimizing (21), we obtain d feature vectors for each view: $z_{xl} = w_{xl}^T X$, $z_{yl} = w_{yl}^T Y$, $l = 1, 2, ..., d$. They are characterized by the following theorem:

Theorem 3.1. *Any two feature vectors z_{xi} and z_{xj} $(i \neq j)$ extracted by multi-view uncorrelated linear discriminant analysis are statistically uncorrelated in view X. And it's the same (statistically uncorrelated) in view Y.*

Proof: It is obvious that the following conditions hold:

$$\begin{array}{l} E[(z_{xi} - Ez_{xi})(z_{xj} - Ez_{xj})] = w_{xi}^T S_t w_{xj} = 0. \\ E[(z_{yi} - Ez_{yi})(z_{yj} - Ez_{yj})] = w_{yi}^T S_t w_{yj} = 0. \end{array} \tag{22}$$

Therefore, the theorem holds. □

Accordingly, the r^{th} discriminant vector pair (w_{xr}, w_{yr}) of matrices X and Y can be obtained in terms of the following theorem.

Theorem 3.2. *The r^{th} discriminant vector pair (w_{xr}, w_{yr}) of matrices X and Y is the eigenvector corresponding to the maximum eigenvalue of the following generalized eigenequation*

$$\begin{bmatrix} P_x & \boldsymbol{0} \\ \boldsymbol{0} & P_y \end{bmatrix} \begin{bmatrix} S_{b_x} & \gamma C_{xy} \\ \gamma C_{yx} & S_{b_y} \end{bmatrix} \begin{bmatrix} w_{xr} \\ w_{yr} \end{bmatrix} = \lambda \begin{bmatrix} S_{t_x} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma S_{t_y} \end{bmatrix} \begin{bmatrix} w_{xr} \\ w_{yr} \end{bmatrix}, \tag{23}$$

where

$$\begin{array}{l} P_x = I - S_{t_x} D_x^T \left(D_x S_{t_x} D_x^T\right)^{-1} D_x, \\ P_y = I - S_{t_y} D_y^T \left(D_y S_{t_y} D_y^T\right)^{-1} D_y, \\ D_x = \left[w_{x1}, w_{x2}, \cdots, w_{x(r-1)}\right]^T, \\ D_y = \left[w_{y1}, w_{y2}, \cdots, w_{y(r-1)}\right]^T, \\ I = diag(1, 1, ..., 1). \end{array} \tag{24}$$

Proof: Since $w_{xr}^T S_{t_x} w_{xr} + \sigma w_{yr}^T S_{t_y} w_{yr} = 1$ and $w_{xr}^T S_{t_x} w_{xj} = w_{yr}^T S_{t_y} w_{yj} = 0$, we construct the corresponding Lagrangian function of (21) in terms of Lagrangian multipliers λ, α_j and β_j

$$\begin{array}{ll} L(w_{xr}, w_{yr}) = & w_{xr}^T S_{b_x} w_{xr} + w_{yr}^T S_{b_y} w_{yr} + 2\gamma w_{xr}^T C_{xy} w_{yr} \\ & -\lambda\left(w_{xr}^T S_{t_x} w_{xr} + \sigma w_{yr}^T S_{t_y} w_{yr} - 1\right) \\ & - \sum_{j=1}^{r-1} 2\alpha_j w_{xr}^T S_{t_x} w_{xj} \\ & - \sum_{j=1}^{r-1} 2\beta_j w_{yr}^T S_{t_y} w_{yj}. \end{array} \tag{25}$$

Taking its derivatives with respect to w_{xr} and w_{yr} to be zero, we have

$$S_{b_x} w_{xr} + \gamma C_{xy} w_{yr} - \lambda S_{t_x} w_{xr} - \sum_{j=1}^{r-1} \alpha_j S_{t_x} w_{xj} = 0, \tag{26}$$

$$S_{b_y} w_{yr} + \gamma C_{yx} w_{xr} - \lambda\sigma S_{t_y} w_{yr} - \sum_{j=1}^{r-1} \beta_j S_{t_y} w_{yj} = 0. \tag{27}$$

Multiplying the left-hand side of (26) and (27) by w_{xr}^T and w_{yr}^T respectively, we obtain

$$2\lambda = w_{xr}^T S_{b_x} w_{xr} + w_{yr}^T S_{b_y} w_{yr} + 2\gamma w_{xr}^T C_{xy} w_{yr}, \tag{28}$$

which means 2λ is equal to the value of the objective function in (21).

Multiplying the left-hand side of (26) by w_{xi}^T, we obtain a set of $r-1$ expressions

$$\begin{array}{l} w_{xi}^T S_{b_x} w_{xr} + \gamma w_{xi}^T C_{xy} w_{yr} - \sum_{j=1}^{r-1} \alpha_j w_{xi}^T S_{t_x} w_{xj} = 0 \\ \\ (i = 1, 2, ..., r-1), \end{array} \tag{29}$$

which can be expressed in another form

$$\begin{bmatrix} w_{x1}^T \\ w_{x2}^T \\ \vdots \\ w_{x(r-1)}^T \end{bmatrix} S_{b_x} w_{xr} + \gamma \begin{bmatrix} w_{x1}^T \\ w_{x2}^T \\ \vdots \\ w_{x(r-1)}^T \end{bmatrix} C_{xy} w_{yr} - \begin{bmatrix} w_{x1}^T \\ w_{x2}^T \\ \vdots \\ w_{x(r-1)}^T \end{bmatrix} S_{tx} \begin{bmatrix} w_{x1}^T \\ w_{x2}^T \\ \vdots \\ w_{x(r-1)}^T \end{bmatrix}^T \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{r-1} \end{bmatrix} = 0. \tag{30}$$

Let

$$\begin{aligned} \alpha &= \left[\alpha_1, \alpha_2, \cdots, \alpha_{r-1}\right]^T, \\ D_x &= \left[w_{x1}, w_{x2}, \cdots, w_{x(r-1)}\right]^T, \end{aligned} \tag{31}$$

so that (29) can be represented in a single matrix relation

$$D_x S_{b_x} w_{xr} + \gamma D_x C_{xy} w_{yr} = D_x S_{t_x} D_x^T \alpha. \tag{32}$$

Thus we obtain

$$\alpha = \left(D_x S_{t_x} D_x^T\right)^{-1} \left(D_x S_{b_x} w_{xr} + \gamma D_x C_{xy} w_{yr}\right). \tag{33}$$

Symmetrically, let

$$\begin{aligned} \beta &= \left[\beta_1, \beta_2, \cdots, \beta_{r-1}\right]^T, \\ D_y &= \left[w_{y1}, w_{y2}, \cdots, w_{y(r-1)}\right]^T, \end{aligned} \tag{34}$$

then we get

$$\beta = \left(D_y S_{t_y} D_y^T\right)^{-1} \left(D_y S_{b_y} w_{yr} + \gamma D_y C_{yx} w_{xr}\right). \tag{35}$$

Using (31), (26) can be rewritten as

$$S_{b_x} w_{xr} + \gamma C_{xy} w_{yr} - \lambda S_{t_x} w_{xr} - S_{t_x} D_x^T \alpha = 0. \tag{36}$$

Substituting (33) into (36), we have

$$\begin{aligned} &\left[I - S_{t_x} D_x^T \left(D_x S_{t_x} D_x^T\right)^{-1} D_x\right] \left(S_{b_x} w_{xr} + \gamma C_{xy} w_{yr}\right) \\ &= \lambda S_{t_x} w_{xr}. \end{aligned} \tag{37}$$

Analogously, from (27) and (35) we have

$$\begin{aligned} &\left[I - S_{t_y} D_y^T \left(D_y S_{t_y} D_y^T\right)^{-1} D_y\right] \left(S_{b_y} w_{yr} + \gamma C_{yx} w_{xr}\right) \\ &= \lambda \sigma S_{t_y} w_{yr}. \end{aligned} \tag{38}$$

Let

$$\begin{aligned} P_x &= I - S_{t_x} D_x^T \left(D_x S_{t_x} D_x^T\right)^{-1} D_x, \\ P_y &= I - S_{t_y} D_y^T \left(D_y S_{t_y} D_y^T\right)^{-1} D_y. \end{aligned} \tag{39}$$

Then we derive the final generalized eigenvalue solution

$$\begin{bmatrix} P_x & \mathbf{0} \\ \mathbf{0} & P_y \end{bmatrix} \begin{bmatrix} S_{b_x} & \gamma C_{xy} \\ \gamma C_{yx} & S_{b_y} \end{bmatrix} \begin{bmatrix} w_{xr} \\ w_{yr} \end{bmatrix} = \lambda \begin{bmatrix} S_{t_x} & \mathbf{0} \\ \mathbf{0} & \sigma S_{t_y} \end{bmatrix} \begin{bmatrix} w_{xr} \\ w_{yr} \end{bmatrix}. \tag{40}$$

□

With d obtained vector pairs (w_{xl}, w_{yl}), $l = 1, 2, ..., d$ after d iterations, let $W_x = [w_{x1}, w_{x2}, ..., w_{xd}]$, $W_y = [w_{y1}, w_{y2}, ..., w_{yd}]$. The combined feature extraction can be obtained according to the following two strategies [25]:

$$I)\ Z = \begin{bmatrix} W_x & \mathbf{0} \\ \mathbf{0} & W_y \end{bmatrix}^T \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{41}$$

$$II)\ Z = \begin{bmatrix} W_x \\ W_y \end{bmatrix}^T \begin{bmatrix} X \\ Y \end{bmatrix}, \tag{42}$$

with d subjecting to the constraints $1 \leq d \leq min(p, q, m)$. Both of them are applicable. In our experiments, we apply the first strategy to fuse extracted features. In addition, since our closed-form solutions are solved by generalized eigenvalue decomposition, to avoid the singularity problem, a regularizer (a multiplication of an identity matrix) [26] is added in our experiments. The main procedure is given in Algorithm 1.

Algorithm 1 Multi-view uncorrelated linear discriminant analysis

Require:
Training data X, Y;
Dimension of the transformed feature space d;
Parameter λ.
Ensure:
Transformed data Z.
1: Construct matrices C_{xy}, S_{b_x}, S_{b_y}, S_{t_x}, S_{t_y} as in (2),(9).
2: $\sigma \leftarrow \frac{tr(S_{t_x})}{tr(S_{t_y})}$.
3: Initialize D_x and D_y to be empty matrices.
4: **for** $r = 1$ **to** d **do**
5: Construct matrices P_x, P_y as in (39);
6: Obtain the r^{th} vector pair (w_{xr}, w_{yr}) by solving (40);
7: Set $D_x = [D_x, w_{xr}]$ (append w_{xr} to D_x as the last column), $D_y = [D_y, w_{yr}]$ (append w_{yr} to D_y as the last column).
8: **end for**
9: $W_x \leftarrow D_x$, $W_y \leftarrow D_y$.
10: Extract features according to (41).
11: **return** Z.

3.3 Modifications of MLDA and MULDA

MLDA utilizes the principle of CCA to exploit the information between views. Through optimizing the objective of CCA, the extracted feature vectors can preserve maximum inter-view correlation in the transformed common subspace. DCCA is an effective supervised feature extraction method for multi-view learning, which can exploit discriminant information between views. Inspired by the fact that DCCA has a similar optimization objective like CCA, we replace $C_{xy} = XY^T$ with ${C_{xy}}' = XAY^T$ in (21), where A is formulated according to (15). The resultant method is called MLDA-m. MULDA can also be extended to MULDA-m with this modification to preserve both inter-view and intra-view class structures.

Discriminant information and correlation information between views are very important in multi-view feature extraction. It is worthwhile to discuss which one is more powerful. We will compare the classification performance of these two types of methods in our experiments.

3.4 The Time Complexity of the Above Linear Feature Extraction Algorithms

In this section, we summarize the time complexity of the above linear feature extraction algorithms in Table 1.

TABLE 1
The Time Complexity of the Linear Feature Extraction Algorithms.

Method	Time complexity
CCA	O($(p+q)^3$)
DCCA	O($(p+q)^3$)
MLDA	O($(p+q)^3$)
MLDA-m	O($(p+q)^3$)
MULDA	O($d(p+q)^3$)
MULDA-m	O($d(p+q)^3$)

4 Kernel Multi-view Uncorrelated Discriminant Analysis

The nonlinear extension of feature extraction methods can improve performance, especially when data have weak linear separability. Kernel methods are suitable to achieve this kind of extension. The main idea in the kernel method is to map an input feature space into a high dimensional feature space, in which a linear problem is solved [27]. The construction of the kernel operator K allows us to solve the original nonlinear problem in a linear way without knowing the implicit nonlinear mapping function. Since MULDA is a linear method, motivated by the properties of the kernel method, we extend it to new nonlinear methods by using kernel-based learning techniques.

In this section, first we introduce the kernel-based version of MLDA, which is called kernel multi-view discriminant analysis [18]. Then KMUDA which is the nonlinear extension of MULDA is proposed with some related theorems and proofs presented. Furthermore, the modification mentioned before is also applied to KMDA and KMUDA. Finally, we provide the time complexity analysis of the nonlinear feature extraction algorithms.

4.1 Kernel Multi-view Discriminant Analysis

Suppose that matrices X and Y are mapped into high dimensional feature spaces as $\phi_x(X) = [\phi_x(x_1), ..., \phi_x(x_n)]$ and $\phi_y(Y) = [\phi_y(y_1), ..., \phi_y(y_n)]$. Using the dual representations and kernel matrices, KMDA can be expressed as

$$\begin{aligned} \max_{a,b} \quad & a^T\phi_x(X)^T\phi_x(X)W\phi_x(X)^T\phi_x(X)a \\ & +b^T\phi_y(Y)^T\phi_y(Y)W\phi_y(Y)^T\phi_y(Y)b \\ & +2\gamma a^T\phi_x(X)^T\phi_x(X)\phi_y(Y)^T\phi_y(Y)b \\ \text{s.t.} \quad & a^T\phi_x(X)^T\phi_x(X)\phi_x(X)^T\phi_x(X)a \\ & +\sigma b^T\phi_y(Y)^T\phi_y(Y)\phi_y(Y)^T\phi_y(Y)b = 1, \end{aligned} \tag{43}$$

where $\sigma = \frac{tr\left(\phi_x(X)^T\phi_x(X)\phi_x(X)^T\phi_x(X)\right)}{tr(\phi_y(Y)^T\phi_y(Y)\phi_y(Y)^T\phi_y(Y))}$ and W is the same as the one in (9). Let $K_x = \phi_x(X)^T\phi_x(X)$ and $K_y = \phi_y(Y)^T\phi_y(Y)$ be the kernel matrices corresponding to these two expressions. Substituting them into (43) results in

$$\begin{aligned} \max_{a,b} \quad & a^TK_xWK_xa + b^TK_yWK_yb + 2\gamma a^TK_xK_yb \\ \text{s.t.} \quad & a^TK_xK_xa + \sigma b^TK_yK_yb = 1, \end{aligned} \tag{44}$$

where $\sigma = \frac{tr(K_xK_x)}{tr(K_yK_y)}$. Using the Lagrangian multiplier technique, this optimization problem can be solved as

$$\begin{bmatrix} K_xWK_x & \gamma K_xK_y \\ \gamma K_yK_x & K_yWK_y \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \lambda \begin{bmatrix} K_xK_x & \mathbf{0} \\ \mathbf{0} & K_yK_y \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}. \tag{45}$$

4.2 Kernel Multi-view Uncorrelated Discriminant Analysis

MULDA may not extract useful uncorrelated feature vectors when dealing with linearly inseparable problems. In the transformed high dimensional kernel spaces, we propose a new method called KMUDA, which aims to exploit not only discriminant but also uncorrelated feature vectors from these two mapped views.

Assuming we have $r-1$ vector pairs $(w_{xj}^\phi, w_{yj}^\phi)$, $j = 1, 2, ..., r-1$, we can express these vector pairs with dual representations: $w_{xj}^\phi = \phi_x(X)a_j$, $w_{yj}^\phi = \phi_y(Y)b_j$, $j = 1, 2, ..., r-1$ similar to (5). KMUDA seeks the r^{th} projection vector pair $(w_{xr}^\phi, w_{yr}^\phi)$ for the mapped matrices $\phi(X)$ and $\phi(Y)$ with the following constraints imposed

$$w_{xr}^{\phi T}S_{tx}^\phi w_{xj}^\phi = w_{yr}^{\phi T}S_{ty}^\phi w_{yj}^\phi = 0. \tag{46}$$

Note that the first vector pair $(w_{x1}^\phi, w_{y1}^\phi)$ is solved by (45) corresponding to the maximum eigenvalue.

Using the dual representations and kernel matrices, the optimization problem of KMUDA is expressed as

$$\begin{aligned} \max_{a_r,b_r} \quad & a_r^TK_xWK_xa_r + b_r^TK_yWK_yb_r \\ & +2\gamma a_r^TK_xK_yb_r \\ \text{s.t.} \quad & a_r^TK_xK_xa_r + \sigma b_r^TK_yK_yb_r = 1, \\ & a_r^TK_xK_xa_j = b_r^TK_yK_yb_j = 0 \\ & (j = 1, 2, ..., r-1). \end{aligned} \tag{47}$$

Once the vector pairs (a_l, b_l), $l = 1, 2, ..., d$ are obtained, we use the following transformation to extract features from the mapped view $\phi(X)$:

$$Z_x^\phi = \begin{bmatrix} z_{x1}^\phi \\ z_{x2}^\phi \\ \vdots \\ z_{xd}^\phi \end{bmatrix} = \begin{bmatrix} w_{x1}^{\phi T} \\ w_{x2}^{\phi T} \\ \vdots \\ w_{xd}^{\phi T} \end{bmatrix} \phi(X) = \begin{bmatrix} a_1^T \\ a_2^T \\ \vdots \\ a_d^T \end{bmatrix} \begin{bmatrix} k(x_1, X) \\ k(x_2, X) \\ \vdots \\ k(x_n, X) \end{bmatrix}, \tag{48}$$

where $k(x_i, X) = \phi_x(x_i)^T\phi_x(X)$. Analogously, for the mapped view $\phi(Y)$, we have the transformed feature matrix Z_y^ϕ.

From (48) we find that since we don't know the explicit nonlinear mapping function ϕ_x and ϕ_y, it is difficult to obtain uncorrelated discriminant vector pairs (w_{xl}, w_{yl}), $l = 1, 2, ..., d$ directly for each view. However, it is very flexible to get (a_l, b_l), $l = 1, 2, ..., d$ by utilizing the kernel function. Therefore we call a_l and b_l , $l = 1, 2, ..., d$ as pseudo-discriminant vectors for convenience.

As discussed in the last section, it is straightforward to obtain the following theorem.

Corollary 4.1. *Any two feature vectors z_{xi}^ϕ and z_{xj}^ϕ $(i \neq j)$ extracted by kernel multi-view uncorrelated discriminant analysis are statistically uncorrelated in the mapped view $\phi(X)$.*

And it's the same (statistically uncorrelated) in the mapped view $\phi(Y)$.

Proof: It is obvious that the following conditions hold:

$$\begin{aligned} E[(z_{xi}^{\phi} - Ez_{xi}^{\phi})(z_{xj}^{\phi} - Ez_{xj}^{\phi})] = w_{xi}^{\phi T} S_t^{\phi} w_{xj}^{\phi} = 0. \\ E[(z_{yi}^{\phi} - Ez_{yi}^{\phi})(z_{yj}^{\phi} - Ez_{yj}^{\phi})] = w_{yi}^{\phi T} S_t^{\phi} w_{yj}^{\phi} = 0. \end{aligned} \tag{49}$$

Therefore, the corollary holds. □

Additionally, we can obtain the r^{th} pseudo-discriminant vector pair (a_r, b_r) of the mapped matrices $\phi(X)$ and $\phi(Y)$ according to the following theorem.

Corollary 4.2. *The r^{th} feature vector pair (a_r, b_r) of the mapped matrices $\phi(X)$ and $\phi(Y)$ is the eigenvector corresponding to the maximum eigenvalue of the following equation:*

$$\begin{bmatrix} P_x^{\phi} & \boldsymbol{0} \\ \boldsymbol{0} & P_y^{\phi} \end{bmatrix} \begin{bmatrix} S_{b_x}^{\phi} & \gamma C_{xy}^{\phi} \\ \gamma C_{yx}^{\phi} & S_{b_y}^{\phi} \end{bmatrix} \begin{bmatrix} a_r \\ b_r \end{bmatrix} = \lambda \begin{bmatrix} S_{t_x}^{\phi} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma S_{t_y}^{\phi} \end{bmatrix} \begin{bmatrix} a_r \\ b_r \end{bmatrix}, \tag{50}$$

where

$$\begin{aligned} &S_{b_x}^{\phi} = K_x W K_x,\ S_{b_y}^{\phi} = K_y W K_y,\ S_{t_x}^{\phi} = K_x K_x, \\ &S_{t_y}^{\phi} = K_y K_y,\ C_{xy}^{\phi} = K_x K_y,\ C_{yx}^{\phi} = K_y K_x, \\ &P_x^{\phi} = I - S_{t_x}^{\phi} D_a^T \left(D_a S_{t_x}^{\phi} D_a^T \right)^{-1} D_a, \\ &P_y^{\phi} = I - S_{t_y}^{\phi} D_b^T \left(D_b S_{t_y}^{\phi} D_b^T \right)^{-1} D_b, \\ &D_a = \begin{bmatrix} a_1, & a_2, & \cdots, & a_{r-1} \end{bmatrix}^T, \\ &D_b = \begin{bmatrix} b_1, & b_2, & \cdots, & b_{r-1} \end{bmatrix}^T, \\ &I = diag(1, 1, ..., 1). \end{aligned} \tag{51}$$

Proof: According to (47), we can construct the corresponding Lagrangian function in terms of Lagrangian multipliers λ, α_j and β_j

$$\begin{aligned} L(a_r, b_r) &= a_r^T K_x W K_x a_r + b_r^T K_y W K_y b_r \\ &\quad + 2\gamma a_r^T K_x K_y b_r \\ &\quad - \lambda \left(a_r^T K_x K_x a_r + \sigma b_r^T K_y K_y b_r - 1 \right) \\ &\quad - \textstyle\sum_{j=1}^{r-1} 2\alpha_j a_r^T K_x K_x a_j \\ &\quad - \textstyle\sum_{j=1}^{r-1} 2\beta_j b_r^T K_y K_y b_j \\ &= a_r^T S_{b_x}^{\phi} a_r + b_r^T S_{b_y}^{\phi} b_r + 2\gamma a_r^T C_{xy}^{\phi} b_r \\ &\quad - \lambda \left(a_r^T S_{t_x}^{\phi} a_r + \sigma b_r^T S_{t_y}^{\phi} b_r - 1 \right) \\ &\quad - \textstyle\sum_{j=1}^{r-1} 2\alpha_j a_r^T S_{t_x}^{\phi} a_j \\ &\quad - \textstyle\sum_{j=1}^{r-1} 2\beta_j b_r^T S_{t_y}^{\phi} b_j. \end{aligned} \tag{52}$$

The remaining proof is similar to the one given in the section in which MULDA is introduced. □

When d pseudo-discriminant vector pairs (a_l, b_l), $l = 1, 2, ..., d$ are obtained after d iterations, the feature extraction can be performed in the feature space using the mapped data, following the method given in (48). The main procedure is listed in Algorithm 2.

4.3 Modifications of KMDA and KMUDA

Observing (44), it is obvious that KMDA can be regarded as a combination of KCCA and GDA. The purpose of KMUDA is to extend our former algorithm MULDA to solve nonlinear problems by utilizing kernel-based learning techniques, so that

Algorithm 2 Kernel multi-view uncorrelated linear discriminant analysis

Require:
Training data $\phi(X)$, $\phi(Y)$;
Dimension of the transformed feature space d;
Parameter λ.
Ensure:
Transformed data Z.
1: Construct matrices C_{xy}^{ϕ}, $S_{b_x}^{\phi}$, $S_{b_y}^{\phi}$, $S_{t_x}^{\phi}$, $S_{t_y}^{\phi}$ as in (51).
2: $\sigma \leftarrow \frac{tr(S_{t_x}^{\phi})}{tr(S_{t_y}^{\phi})}$.
3: Initialize D_a and D_a to be empty matrices.
4: **for** $r = 1$ **to** d **do**
5: Construct matrices P_x^{ϕ}, P_y^{ϕ} as in (51);
6: Obtain the r^{th} vector pair (a_r, b_r) by solving (50);
7: Set $D_a = [D_a, a_r]$ (append a_r to D_a as the last column), $D_b = [D_b, b_r]$ (append b_r to D_b as the last column).
8: **end for**
9: $W_x \leftarrow D_a$, $W_y \leftarrow D_b$.
10: Extract features according to (41).
11: **return** Z.

we can extract feature vectors with maximum discrimination in each view and correlation between views from the possibly linearly inseparable two-view data. Furthermore, these feature vectors extracted from the mapped datasets will be mutually uncorrelated in each view. Similar to the last section, which has similar optimization expressions compared (17) with (6), we can replace the KCCA part with KDCCA in KMDA and KMUDA. In this case, the feature vectors extracted by this modification will contain minimum within-class distance and maximum between-class distance for both intra-view and inter-view. In other words, the class structure information can be preserved not only in each view but also between views while the redundant information is removed in the common subspace. We also make experiments to study the effect of this modification and name these two methods as KMDA-m and KMUDA-m.

4.4 The Time Complexity of the Above Nonlinear Feature Extraction Algorithms

In this section, we summarize the time complexity of the above nonlinear feature extraction algorithms in Table 2.

TABLE 2
The Time Complexity of the Nonlinear Feature Extraction Algorithms.

Method	Time complexity
KCCA	O(n^3)
KDCCA	O(n^3)
KMDA	O(n^3)
KMDA-m	O(n^3)
KMUDA	O(dn^3)
KMUDA-m	O(dn^3)

5 EXPERIMENTS

In this section, we evaluate the performance of our methods for extracting features from two-view data for classification. Experiments are performed on two types of datasets: (a) Multiple-feature dataset for handwritten digit classification, (b) PIE human face dataset for face recognition.

The methods used for comparison are the following:

kNN: k-nearest-neighbor classifier with $k = 3$ is applied directly on the original two-view data;

KkNN: k-nearest-neighbor classifier with $k = 3$ is applied on the mapped two-view data in the kernel space;

CCA: Canonical correlation analysis to extract features from the two-view data for classification;

KCCA: Kernel canonical correlation analysis [17];

DCCA: Discriminant canonical correlation analysis [5];

KDCCA: Kernelized discriminant canonical correlation analysis [20];

MvDA: Multi-view discriminant analysis [9];

MLDA: Multi-view linear discriminant analysis;

MULDA: Multi-view uncorrelated linear discriminant analysis;

MLDA-m: Multi-view linear discriminant analysis with modifications;

MULDA-m: Multi-view uncorrelated linear discriminant analysis with modifications;

KMDA: Kernel multi-view discriminant analysis;

KMUDA: Kernel multi-view uncorrelated discriminant analysis;

KMDA-m: Kernel multi-view discriminant analysis with modifications;

KMUDA-m: Kernel multi-view uncorrelated discriminant analysis with modifications.

After using the feature extraction methods, the kNN classifiers with k=3, k=5 and k=7 are applied for classification. For all the kernel-based methods, the commonly used Gaussian kernel is employed, and the kernel width parameters are optimized among $[2^{-3}, 2^{-2}, ..., 2^4]$ multiplying the mean squared distances between examples. In addition, the tuning parameter γ in MLDA, MLDA-m, MULDA and MULDA-m is optimized among $[1, 5, 10, 15, 20]$, while in KMDA, KMDA-m, KMUDA and KMUDA-m this parameter is set to 10. The average classification accuracies and standard deviations are recorded during 10 random experiments. More details are reported in the following subsections.

5.1 Multiple-Feature Dataset

In this subsection, we evaluate the effectiveness of our methods on handwritten digit classification. First we introduce the dataset. Then the effect of the number of reduced dimensions on the classification performance of MULDA is studied. At last, we compare all the methods listed above in terms of classification accuracies.

5.1.1 Dataset

The multiple-feature database is available from the UCI repository. It is composed of features of handwritten digits ('0'-'9') extracted from a collection of Dutch utility maps. 200 examples per class (for a total of 2,000 examples) have been digitized in binary images. Six sets of features, which respectively describe different views of the digits are included. The six feature sets and number of attributes in each set are listed as follows: 1) Fourier coefficients of the character shapes (**FOU**,76); 2) Profile correlations (**FAC**,216); 3) Karhunen-Love coefficients (**KAR**,64); 4) Pixel averages in 2 x 3 windows (**PIX**,240); 5) Zernike moments (**ZER**,47); 6) Morphological features (**MOR**,6).

Any two of them are picked out to construct view X and view Y, so that there are in total 15 pairs of different combinations and each combination forms a two-view dataset. For each class, we randomly pick out 100 pairs of feature vectors for training, and the remaining for test. In the implementation of the methods, five-fold cross-validation is used to select the optimal parameters.

5.1.2 Effect of the number of reduced dimensions on MULDA

In this experiment, we study the effect of the number of reduced dimensions on the classification performance of MULDA. The dimension of the common subspace is restricted to be $\tilde{d}$ by keeping the first $\tilde{d}$ projection vectors only, where $1 \leq \tilde{d} \leq min(p, q, m)$. In Fig.1 we show the classification results on the combination of **PIX** and **ZER**, where the horizontal axis represents the reduced dimensions and the vertical axis represents the classification accuracy. It can be observed that the accuracy increases monotonically as the number of reduced dimensions increases, until $\tilde{d} = m - 1$ is reached. This observation is consistent with the theory in [3], that is, the optimal dimensionality of the extracted feature space is $m - 1$. Since results on the other two-view datasets are similar, we do not present them here. Based on these observations, except the two-view datasets in which feature set **MOR** is included, the reduced dimensions of MULDA are set to $m - 1$. The reduced dimensions for the exceptions are set to be the dimension of feature vectors belonging to **MOR**.

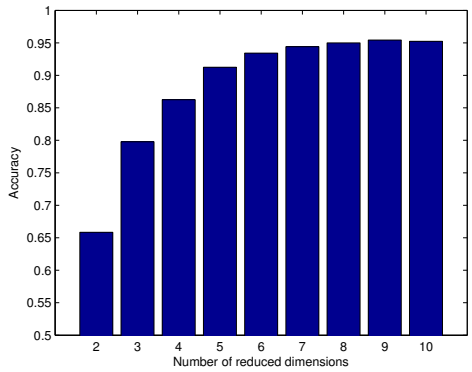


Fig. 1. Effect of the number of reduced dimensions on the classification performance of MULDA for the combination of **PIX** and **ZER**.

TABLE 3
Classification Accuracy and Standard Deviation (%) on the Multiple-Feature Database for the Linear Case and t-test Results ($k = 3$).

X	Y	kNN	CCA	DCCA	MvDA	MLDA	MLDA-m	MULDA	MULDA-m
FOU	**KAR**	95.45±0.59	84.36±0.68	89.89±1.12	91.76±0.70	**97.53**±0.31	96.88±0.44	97.29±0.47	96.64±0.40
FOU	**FAC**	96.74±0.36	88.02±0.86	91.75±0.97	78.08±1.09	**97.91**±0.43	97.30±0.44	97.76±0.42	97.53±0.33
FOU	**PIX**	96.91±0.37	84.81±0.52	90.12±0.85	90.43±0.74	**97.52**±0.49	97.11±0.44	97.12±0.45	97.13±0.31
FOU	**ZER**	83.60±0.56	80.45±1.04	83.34±0.64	70.87±1.41	**85.67**±0.56	85.51±0.65	85.37±0.76	85.58±0.92
FOU	**MOR**	80.65±0.70	76.34±1.09	**83.35**±1.09	70.64±1.08	82.98±0.92	83.19±0.88	82.47±0.77	83.18±0.90
KAR	**FAC**	96.44±0.43	88.93±0.85	95.78±0.46	80.12±1.40	96.44±0.84	97.12±0.35	**97.24**±0.52	97.12±0.35
KAR	**PIX**	96.40±0.33	87.05±0.91	93.89±0.61	92.65±0.43	**97.23**±0.46	95.25±0.60	95.91±0.39	94.87±0.48
KAR	**ZER**	95.25±0.41	69.49±0.62	88.36±1.44	84.04±0.81	95.91±0.39	**96.45**±0.30	96.16±0.42	96.31±0.56
KAR	**MOR**	95.50±0.38	80.70±1.33	91.99±1.41	87.16±1.24	**96.65**±0.33	94.27±1.46	96.58±0.54	94.26±0.15
FAC	**PIX**	**96.94**±0.41	85.19±0.76	95.86±0.68	81.95±1.47	96.86±0.59	96.89±0.70	96.69±0.52	96.81±0.59
FAC	**ZER**	96.35±0.38	74.77±0.97	88.15±1.12	92.84±1.68	97.02±0.45	**97.41**±0.56	97.10±0.43	97.25±0.68
FAC	**MOR**	97.23±0.34	82.99±1.86	93.02±1.03	87.24±1.69	**97.69**±0.49	94.20±0.19	97.29±0.80	94.16±1.90
PIX	**ZER**	**96.64**±0.28	66.40±1.35	87.82±1.63	79.60±1.58	95.72±0.45	95.59±0.56	95.19±0.66	95.56±0.45
PIX	**MOR**	**96.96**±0.28	77.61±0.82	91.84±1.66	80.60±2.17	96.35±0.51	92.38±1.70	96.09±0.67	92.36±1.70
ZER	**MOR**	82.02±0.11	72.80±1.57	82.76±1.03	70.17±1.55	82.93±0.65	**83.24**±0.91	81.88±0.73	83.22±0.92
t-test		1	1	1	1	/	0	1	1

TABLE 4
Classification Accuracy and Standard Deviation (%) on the Multiple-Feature Database for the Linear Case and t-test Results ($k = 5$).

X	Y	kNN	CCA	DCCA	MvDA	MLDA	MLDA-m	MULDA	MULDA-m
FOU	**KAR**	95.60±0.39	85.06 ±0.63	90.26±0.75	92.08±0.54	**97.68**±0.39	96.91±0.26	97.21±0.35	96.68±0.61
FOU	**FAC**	97.01±0.34	87.82 ±1.05	91.73±1.20	77.17±1.42	**97.86**±0.41	91.73±1.20	97.65±0.52	97.44±0.38
FOU	**PIX**	96.94±0.40	84.82±0.65	90.28±0.54	91.14±0.32	**97.43**±0.49	97.14±0.42	96.97±0.38	97.01±0.47
FOU	**ZER**	83.84±0.39	75.78±1.07	83.91±0.75	70.25±1.11	85.89±0.80	85.76±0.81	85.76±0.75	**86.00**±0.52
FOU	**MOR**	81.05±1.05	76.55±1.42	83.49 ±0.95	70.14±1.40	**83.66**±0.95	83.43±0.59	82.55±0.84	83.39±0.58
KAR	**FAC**	96.33±0.40	88.81±0.80	96.00±0.71	81.63±1.37	**97.28**±0.56	97.27±0.52	97.21±0.54	97.27±0.52
KAR	**PIX**	**96.39**±0.28	87.57±1.57	94.11±0.85	92.98±0.55	95.91±0.40	95.48±0.62	95.92±0.40	95.30±0.74
KAR	**ZER**	95.17±0.54	71.04±1.04	88.68±1.39	85.35±1.19	96.27±0.54	**96.40**±0.50	96.08±0.56	96.34±0.49
KAR	**MOR**	95.39±0.38	81.34 ±0.98	92.15±1.28	87.72±1.61	96.57±0.55	94.42±1.36	**96.60**±0.59	94.40±1.36
FAC	**PIX**	96.91±0.52	85.43±1.21	95.86±0.67	83.57±1.48	96.87±0.54	96.89±0.54	96.71±0.49	**97.09**±0.51
FAC	**ZER**	96.49±0.36	75.78±1.07	88.57±1.39	93.08±0.76	96.96±0.44	**97.43**±0.59	97.15±0.44	97.42±0.49
FAC	**MOR**	97.11±0.41	83.50±1.16	93.19±0.13	87.97±2.34	**97.72**±0.52	94.30±1.94	97.36±0.74	94.32±1.91
PIX	**ZER**	**96.60**±0.21	68.17±0.92	93.99±1.31	81.44±1.30	95.54±0.49	95.70±0.56	95.22±0.62	95.72±0.66
PIX	**MOR**	**96.90**±0.31	78.12±0.63	91.78±0.16	82.13±2.41	96.36±0.57	92.37±1.78	96.19±0.64	92.35±1.77
ZER	**MOR**	82.10±0.76	74.01±1.01	82.77±0.97	71.51±2.62	83.15±0.54	**83.38**±0.97	82.70±0.76	83.37±0.53
t-test		1	1	1	1	/	1	1	0

TABLE 5
Classification Accuracy and Standard Deviation (%) on the Multiple-Feature Database for the Linear Case and t-test Results ($k = 7$).

X	Y	kNN	CCA	DCCA	MvDA	MLDA	MLDA-m	MULDA	MULDA-m
FOU	**KAR**	95.49±0.34	85.06±0.63	90.38±0.90	92.00±0.76	**97.58**±0.37	96.95±0.48	97.06 ±0.30	96.78±0.22
FOU	**FAC**	96.90±0.49	87.82±1.05	91.88±1.20	75.24±1.43	**97.86**±0.40	97.46±0.33	97.71±0.41	97.42±0.60
FOU	**PIX**	96.81±0.22	84.82±0.65	90.30±0.90	90.82±0.67	**97.23**±0.50	97.13±0.34	96.83±0.40	96.98±0.32
FOU	**ZER**	83.87±0.79	80.80±0.83	84.12±0.88	68.33±1.31	**86.17**±0.67	85.61±0.72	85.54±0.64	85.79±0.70
FOU	**MOR**	80.96±0.91	76.55±1.42	83.28 ±0.92	68.43±1.08	83.50±0.83	83.75±0.66	82.87±0.47	**83.78**±0.63
KAR	**FAC**	96.16±0.46	88.64±0.92	95.84±0.84	81.31±1.14	**97.18**±0.56	97.10±0.67	97.09±0.58	97.10±0.67
KAR	**PIX**	**96.23**±0.32	87.69±0.79	94.25±0.77	92.55±0.79	95.88±0.49	95.33±0.54	95.91±0.49	95.28±0.67
KAR	**ZER**	94.95±0.34	71.71±1.00	88.70±1.33	85.62±1.46	96.25±0.54	**96.34**±0.36	95.89±0.54	96.31±0.35
KAR	**MOR**	95.10±0.46	81.43±1.10	91.91±1.53	87.83±1.69	**96.67**±0.55	94.55±1.45	96.57±0.70	94.52±1.46
FAC	**PIX**	**96.89**±0.45	85.07±0.93	95.98±0.66	82.55±1.85	96.67±0.55	96.89±0.55	96.60±0.61	96.84±0.59
FAC	**ZER**	96.41±0.41	76.31±0.94	88.63±1.44	92.92±1.05	97.05±0.64	**97.47**±0.39	97.04±0.47	97.43±0.48
FAC	**MOR**	96.92±0.46	83.64±0.98	93.29±1.08	88.12±2.20	**97.78**±0.53	94.48±1.58	97.36±0.72	94.47±1.60
PIX	**ZER**	**96.57**±0.34	68.58±1.30	88.53±1.25	81.59±1.47	95.47±0.69	95.63±0.74	95.09±0.64	95.63±0.48
PIX	**MOR**	**96.80**±0.27	78.54±0.84	91.79±1.77	81.26±2.33	96.35±0.60	92.78±1.65	96.19±0.73	92.77±1.64
ZER	**MOR**	82.26±0.37	74.41±1.11	83.04±0.57	71.54±1.60	83.43±0.67	**83.72**±1.03	82.81±0.67	**83.72**±0.94
t-test		1	1	1	1	/	0	1	0

TABLE 6
Classification Accuracy and Standard Deviation (%) on the Multiple-Feature Database for the Nonlinear Case and t-test Results ($k = 3$).

X	Y	KCCA	KDCCA	KMDA	KMDA-m	KMUDA	KMUDA-m
FOU	**KAR**	86.32±1.36	93.78±1.09	96.74±0.27	86.60±1.22	96.74±0.27	**98.58**±0.31
FOU	**FAC**	88.89±0.85	94.74±0.65	96.89±0.62	89.83±0.97	96.89±0.62	**98.64**±0.45
FOU	**PIX**	88.08±0.66	94.29±0.49	96.86±0.44	88.75±1.08	96.86±0.44	**98.61**±0.28
FOU	**ZER**	81.27±0.97	86.17±0.82	**87.76**±0.88	85.26±0.75	**87.76**±0.88	87.53±0.77
FOU	**MOR**	80.71±1.30	82.21±0.70	79.20±1.06	79.15±0.91	82.85±0.86	**85.42**±0.45
KAR	**FAC**	88.48±0.80	97.69±0.52	**98.50**±0.38	95.86±0.77	**98.50**±0.38	98.45±0.38
KAR	**PIX**	87.05±1.38	97.42±0.38	98.08±0.33	95.86±0.68	98.08±0.33	**98.16**±1.34
KAR	**ZER**	73.69±1.15	93.04±0.54	94.33±0.83	86.19±1.03	94.33±0.83	**98.05**±0.30
KAR	**MOR**	85.85±0.75	90.27±1.21	84.36±1.36	81.47±1.76	95.27±0.82	**98.12**±0.29
FAC	**PIX**	87.67±0.97	97.68±0.51	98.41±0.28	**98.42**±0.27	98.41±0.28	**98.42**±0.27
FAC	**ZER**	80.03±0.82	93.81±0.50	94.52±0.79	98.11±0.33	94.52±0.79	**98.40**±0.29
FAC	**MOR**	87.39±0.77	93.28±1.54	83.71±1.35	**98.15**±0.29	93.53±0.16	**98.15**±0.29
PIX	**ZER**	71.03±1.12	92.92±0.55	94.82±0.85	86.05±2.21	94.82±0.85	**98.17**±0.28
PIX	**MOR**	85.96±1.06	91.45±1.58	86.30±1.15	87.58±1.43	92.28±0.87	**98.32**±0.27
ZER	**MOR**	77.68±0.74	83.10±1.02	76.89±0.81	77.40±2.20	81.05±0.34	**84.58**±0.34
t-test		1	1	1	1	1	/

TABLE 7
Classification Accuracy and Standard Deviation (%) on the Multiple-Feature Database for the Nonlinear Case and t-test Results ($k = 5$).

X	Y	KCCA	KDCCA	KMDA	KMDA-m	KMUDA	KMUDA-m
FOU	**KAR**	87.35±1.39	93.75 ± 0.89	96.82 ± 0.36	86.82 ± 1.31	96.82 ± 0.36	**98.58**±0.33
FOU	**FAC**	90.77±0.79	94.97 ± 0.72	97.11 ± 0.36	89.20 ± 1.25	97.11 ± 0.36	**98.53**±0.42
FOU	**PIX**	89.55±1.00	94.33 ± 0.54	96.87 ± 0.42	87.89 ± 1.11	96.87 ± 0.42	**98.67**±0.31
FOU	**ZER**	81.87±0.71	86.71 ± 0.46	**87.52**±0.68	85.59 ± 0.89	87.52 ± 0.68	87.06±0.52
FOU	**MOR**	80.83±1.28	82.14 ± 0.99	79.24 ± 1.42	79.48 ± 1.27	82.66 ± 0.96	**85.21**±0.55
KAR	**FAC**	91.62±0.76	97.69 ± 0.54	98.42±0.27	94.31 ± 0.94	98.42 ± 0.27	**98.46**±0.35
KAR	**PIX**	90.22±1.22	97.36 ± 0.40	98.01 ± 0.31	96.04 ± 0.70	98.01 ± 0.31	**98.14**±0.28
KAR	**ZER**	79.37±1.41	93.24 ± 0.50	94.39 ± 0.94	85.40 ± 1.04	94.39 ± 0.94	**97.99**±0.33
KAR	**MOR**	87.59±0.99	90.17 ± 1.18	84.87 ± 1.15	81.61 ± 2.39	95.37 ± 0.58	**98.19**±0.48
FAC	**PIX**	89.14±0.81	97.75 ± 0.58	98.37 ± 0.28	95.25 ± 1.02	98.37 ± 0.28	**98.39**±0.27
FAC	**ZER**	84.90±1.17	93.96 ± 0.60	94.64 ± 0.76	88.99 ± 0.92	94.64 ± 0.76	**98.36**±0.40
FAC	**MOR**	90.35±0.96	93.31 ± 1.56	89.34 ± 1.06	88.45 ± 1.33	94.58 ± 0.82	**98.18**±0.42
PIX	**ZER**	73.53±1.07	93.17 ± 0.55	95.11 ± 0.91	86.79 ± 0.90	95.11 ± 0.91	**98.12**±0.21
PIX	**MOR**	86.20±1.03	91.57 ± 1.56	86.70 ± 1.08	88.42 ± 1.73	92.43 ± 0.85	**98.28**±0.38
ZER	**MOR**	78.46±0.86	83.88 ± 0.59	77.50 ± 0.76	78.06 ± 2.42	81.69 ± 0.84	**85.06**±0.42
t-test		1	1	1	1	1	/

TABLE 8
Classification Accuracy and Standard Deviation (%) on the Multiple-Feature Database for the Nonlinear Case and t-test Results ($k = 7$).

X	Y	KCCA	KDCCA	KMDA	KMDA-m	KMUDA	KMUDA-m
FOU	**KAR**	88.13±0.76	93.36 ± 0.77	96.75 ± 0.34	86.66 ± 1.39	96.75 ± 0.34	**98.56**±0.30
FOU	**FAC**	90.70±1.01	94.91 ± 0.65	97.12 ± 0.43	90.23 ± 1.09	97.12 ± 0.43	**98.59**±0.35
FOU	**PIX**	89.35±1.13	94.22 ± 0.58	96.89 ± 0.47	88.19 ± 1.27	96.89 ± 0.47	**98.67**±0.25
FOU	**ZER**	81.97±1.07	86.83 ± 0.68	**87.38**±0.65	86.36 ± 0.80	**87.38**±0.65	86.93 ± 0.59
FOU	**MOR**	80.85±1.17	82.51 ± 0.76	79.10 ± 1.30	79.57 ± 0.89	82.97 ± 1.04	**85.51**±0.47
KAR	**FAC**	91.87±0.59	97.60 ± 0.60	**98.47**±0.32	94.36 ± 0.91	**98.47**±0.32	**98.47**±0.36
KAR	**PIX**	91.04±1.15	97.27 ± 0.32	**98.08** ± 0.29	**98.08** ± 0.35	**98.08** ± 0.29	**98.08**±0.35
KAR	**ZER**	79.75±1.68	93.19 ± 0.43	94.21 ± 0.96	88.71 ± 1.21	94.21 ± 0.96	**98.08**±0.24
KAR	**MOR**	87.24±1.09	90.91 ± 0.74	84.63 ± 0.89	82.38 ± 2.00	95.49 ± 0.57	**98.19**±0.43
FAC	**PIX**	89.11±0.95	97.73 ± 0.53	98.41 ± 0.28	95.76 ± 0.83	98.41 ± 0.28	**98.42**±0.34
FAC	**ZER**	84.78±0.98	94.02 ± 0.51	94.52 ± 0.59	89.05 ± 0.84	94.52 ± 0.59	**98.33**±0.45
FAC	**MOR**	89.99±1.01	93.08 ± 1.63	84.12 ± 1.30	84.79 ± 1.95	93.82 ± 1.75	**98.27**±0.44
PIX	**ZER**	73.72±1.25	93.06 ± 0.51	94.96 ± 1.11	87.02 ± 0.99	94.96 ± 1.11	**98.18**±0.38
PIX	**MOR**	87.23±0.86	91.28 ± 1.58	86.70 ± 1.03	88.29 ± 1.63	92.47 ± 1.04	**98.30**±0.43
ZER	**MOR**	79.06±0.86	84.12 ± 0.65	77.84 ± 0.70	82.15 ± 1.16	82.23 ± 0.85	**84.70**±0.71
t-test		1	1	1	1	1	/

5.1.3 Comparison of classification performance

The methods we compare in this paper can be divided into two categories which are linear methods and nonlinear methods. In the linear case, we compare our methods MLDA-m, MULDA and MULDA-m with kNN, CCA, DCCA, MvDA and MLDA. The reduced dimensions of CCA and DCCA are set to be the same as MULDA. The results are summarized in Tables 3∼5. We further use the t-test method [31] to show the significance of the results. All the other methods are compared with the performance of MLDA with the significance level of 0.05 in Tables 3∼5 (MLDA is selected because its performance appears to be the best). The value 1 represents that the compared two methods have a significant performance difference, while the value 0 represents not.

In the nonlinear case, the methods used to compare with our methods KMDA-m, KMUDA and KMUDA-m are KCCA, KDCCA and KMDA. All the other methods are compared with the performance of KMUDA-m with the significance level of 0.05 (KMUDA-m is selected because its performance appears to be the best). Tables 6∼8 list the classification results of these methods and the results of t-test. Obviously we can observe that KMUDA-m achieves the best accuracy. More discussions of these results can be found in Section 5.3.

5.2 PIE Dataset

In this section, we evaluate the performance of our algorithms on face recognition datasets. The dataset is CMU PIE, which will be introduced below. Then the comparison of all the methods is reported.

5.2.1 Dataset

The CMU PIE database (available at http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html) consists of a collection of face images under varying poses, illuminations and expressions. It contains 68 subjects, each with 13 different poses, 43 different illumination conditions and 4 different expressions. Some preprocessing steps had been done for images in this database [28][29]. Face areas in each image are cropped to the size of 64 by 64 pixels, and then resized to 32 by 32 pixels. Twenty subjects with the frontal pose are chosen for our experiments and each subject has 49 collections. Thus we have a face database with 980 images. PCA is applied to this database with 100 percent of total energy preserved. Finally, the transformed 64 by 64 pixels database is reduced to the dimensionality 42 and it is constructed as view X. The processed 32 by 32 pixels database is reduced to the dimensionality 25 and it is set to be view Y. In each experiment, 20 images of each person are randomly selected for training, and the remaining 29 images are used for test. Four-fold cross-validation is used to estimate parameters.

5.2.2 Comparison of classification performance

In this experiment, we compare the performance of each algorithm on face recognition datasets. Tables 9∼11 show the classification accuracies on the CMU PIE database.

TABLE 9
Classification Accuracy and Standard Deviation (%) on the PIE Database ($k = 3$).

Method	Classification accuracy
kNN	95.67±2.46
CCA	89.50±2.91
DCCA	96.57±1.55
MvDA	95.64±2.21
MLDA	96.67±1.50
MLDA-m	97.32±1.06
MULDA	96.71±1.62
MULDA-m	96.76±1.52
KkNN	97.36±1.24
KCCA	91.86±2.53
KDCCA	97.88±1.22
KMDA	98.36±0.89
KMDA-m	98.10±1.07
KMUDA	98.36±0.89
KMUDA-m	**98.52**±0.80

TABLE 10
Classification Accuracy and Standard Deviation (%) on the PIE Database ($k = 5$).

Method	Classification accuracy
kNN	93.47±3.78
CCA	85.83±4.21
DCCA	96.29 ±1.40
MvDA	94.95±2.42
MLDA	96.02±1.66
MLDA-m	97.05±0.93
MULDA	95.76 ±1.76
MULDA-m	96.12±1.77
KkNN	94.10±2.70
KCCA	89.24±3.29
KDCCA	97.74±1.14
KMDA	**98.97**±0.57
KMDA-m	98.95±0.57
KMUDA	**98.97**±0.57
KMUDA-m	98.95±0.57

TABLE 11
Classification Accuracy and Standard Deviation (%) on the PIE Database ($k = 7$).

Method	classification accuracy
kNN	92.03±3.69
CCA	83.41±3.67
DCCA	96.07±1.55
MvDA	93.48±2.59
MLDA	95.62±1.90
MLDA-m	96.55±1.08
MULDA	95.00±2.21
MULDA-m	95.59±1.92
KkNN	92.69±2.55
KCCA	86.91±2.81
KDCCA	93.96±1.63
KMDA	**98.40**±1.08
KMDA-m	98.38±1.08
KMUDA	**98.40**±1.08
KMUDA-m	98.38±1.08

5.3 Discussions of Experimental Results

From Tables 3∼5, we can observe that the classification performances of our methods MLDA-m, MULDA and MULDA-m are better than the other linear multi-view feature extraction methods except MLDA in most cases. Our methods outperform kNN which applies the classifier directly on the original two-view dataset. Moreover, in those cases that kNN or MLDA is superior, our methods are still very competitive. The results of the linear methods in Tables 9∼11 can further demonstrate the good performance of the proposed methods. In Tables 9∼11, MULDA-m performs better than MULDA. MLDA-m performs best in all the linear methods.

Comparing the results in Tables 3∼5 with the corresponding results in Tables 6∼8, we can first observe that the kernel extension of each linear method can bring improvement in classification performance in most cases. And in Tables 9∼11 the result of each kernel-based method is also better than the corresponding linear method except the results of KDCCA and DCCA in Table 11. This is consistent with the theory that kernel representation offers an alternative solution to increase the power of the linear learning method [17]. From Tables 6∼8 and Tables 9∼11, it is obvious that our methods KMUDA and KMUDA-m are generally better than all the other methods. However, the performance of KMDA-m is not better than KMDA in most cases.

Comparing Table 3 with Table 6, we can observe that MLDA performs better than KMDA in eleven cases, and MLDA-m performs better than KMDA-m in eleven cases. MULDA performs better than KMUDA in ten cases. Comparing Table 4 with Table 7, we can observe that MLDA performs better than KMDA in fourteen cases, and MLDA-m performs better than KMDA-m in fourteen cases. MULDA performs better than KMUDA in ten cases. Comparing Table 5 with Table 9, we can observe that MLDA performs better than KMDA in eleven cases, and MLDA-m performs better than KMDA-m in thirteen cases. MULDA performs better than KMUDA in nine cases. We speculate the reason is that the tuning parameter γ in MLDA, MLDA-m, MULDA and MULDA-m are optimized among $[1, 5, 10, 15, 20]$, while in KMDA, KMDA-m, KMUDA and KMUDA-m this parameter is set to 10 in order to reduce the numbers of tuning parameters.

About our methods, an interesting and obvious result can be observed from Tables 6∼8. KMUDA-m outperforms KMUDA in almost all two-view datasets. For the other cases, the accuracies of KMUDA-m and KMUDA are very close. So we can speculate that KMUDA with modifications can achieve better classification performance than KMUDA. This means that the discriminant information between views contributes more than the correlation information between views when extracting features from the transformed high-dimensional feature space. However, in the linear case, we can just say that MULDA with modifications is as effective as MULDA.

From the t-test results of Table 3, we can observe that the performance difference between MLDA and MLDA-m is insignificant. From the t-test results of Table 4, we can observe that the performance difference between MLDA and MULDA-m is insignificant. From the t-test results of Table 5, we can observe that the performance difference between MLDA and MLDA-m is insignificant and the performance difference between MLDA and MULDA-m is insignificant. We speculate the reason is that class structures between views in MLDA-m and MULDA-m is not very informative on this dataset. From the t-test results of Tables 6∼8, we can observe that KMUDA-m and the other methods are significantly different. We conclude that KMUDA-m can achieve the best classification performance.

6 Conclusion and Future Work

In this paper, we proposed a new method MULDA, which utilizes the principles of CCA and ULDA to take advantage of these two algorithms. By optimizing the objective function, both class structures in each view and correlation information between views can be preserved in the transformed common subspace. To exploit the inter-view discriminant information, we also modified MULDA by replacing the CCA part with DCCA, which is able to utilize the class information between views in the combined feature extraction. Simultaneously, we modified MLDA by replacing the CCA part with DCCA. Additionally, in order to deal with the possibly linearly inseparable problem, we proposed a novel algorithm to combine the kernel method into MULDA called KMUDA. The features extracted by KMUDA can simultaneously maximize the discrimination in each view and correlation between views, which makes it suitable for classification in problems of weak linear separability. Similarly, we proposed another new method, KMUDA with modifications, which aims to preserve class structures not only in each view but also between views. Moreover, owing to the integration of the uncorrelated constraints, the feature vectors extracted by our methods are mutually uncorrelated in the common subspace. Simultaneously, we also extended KMDA to KMDA-m.

To evaluate the effectiveness of our methods, we performed experiments to compare our methods with other related and state-of-the-art methods on handwritten digit classification and face recognition datasets. The experimental results validate that MLDA-m, MULDA and MULDA-m are superior to the other methods in the linear cases except MLDA, and KMUDA and KMUDA-m outperform the other nonlinear methods in most cases. The improvements of accuracies after nonlinear extensions verify that the performance can be increased by using the kernel representation. Moreover, the comparison between MULDA and MULDA-m shows that MULDA-m is as competitive as MULDA, which implies that when dealing with the linear problems, correlation information and discriminant information are both useful for classification. However, it's very interesting that the modification of KMUDA can significantly improve the performance in most cases for the nonlinear problems. Our kernel methods give a significant improvement on the classification performance. So we speculate that preserving class structures between views for feature extraction can provide more powerful information in the nonlinear scenario. In conclusion, our methods perform better than the other related methods for extracting features from two-view data for classification.

In our methods, each feature vector corresponds to a generalized eigenvalue decomposition process. For large and high-dimensional datasets, our algorithm may be computationally expensive, and thus we will try to exploit more efficient closed-form solutions, such as [12]. In the future, it is also interesting to extend the current work to semi-supervised learning [30].

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Project 61370175, and Shanghai Knowledge Service Platform Project (No. ZF1213).

REFERENCES

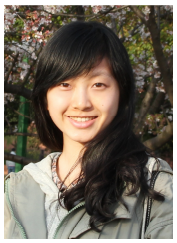
[1] S. Sun, "A survey of multi-view machine learning," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2031-2038, 2013.
[2] H. Hotelling, "Relations between two sets of variates," *Biometrica*, vol. 28, no. 3-4, pp. 321-377, 1936.
[3] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.
[4] H. Wang, X. Lu, Z. Hu and W. Zheng, "Fisher discriminant analysis with L1-Norm," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 828-842, 2014.
[5] T. Sun, S. Chen, J. Yang and P. Shi, "A novel method of combined feature extraction for recognition," *in Proceedings of the International Conference on Data Mining*, 2008, Pisa, Italy, pp. 1043-1048.
[6] T. Diethe, D. R. Hardoon and J. Shawe-Taylor, "Multiview fisher discriminant analysis," *in Proceedings of the Neural Information Processing Systems Workshop on Learning from Multiple Sources*, 2008, Vancouver, Canada, pp. 1-8.
[7] T. Diethe, D. Hardoon and J. Shawe-Taylor, "Constructing nonlinear discriminants from multiple data views," *in Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2010, Barcelona, Spain, pp. 328-343.
[8] Q. Chen and S. Sun, "Hierarchical multi-view fisher discriminant analysis," *in Proceedings of the 16th International Conference on Neural Information Processing*, 2009, Bangkok, Thailand, pp. 289-298.
[9] M. Kan, S. Shan, H. Zhang, S. Lao and X. Chen, "Multi-view discriminant analysis," *in Proceedings of the European Conference on Computer Vision*, 2012, Firenze, Italy, pp. 808-821.
[10] M. Yang and S. Sun, "Multi-view uncorrelated linear discriminant analysis with applications to handwritten digit recognition," *in Proceedings of the International Joint Conference on Neural Networks*, 2014, Beijing, China, pp. 4175-4181.
[11] Z. Jin, J Y. Yang, Z S. Hu and Z. Lou, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, vol. 34, no. 7, pp. 1405-1416, 2001.
[12] J. Ye, T. Li, T. Xiong and R. Janardan, "Using uncorrelated discriminant analysis for tissue classification with gene expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 4, pp. 181-190, 2004.
[13] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *Journal of Machine Learning Research*, vol. 6, no. 4, pp. 483-502, 2005.
[14] J. Ye, R. Janardan, Q. Li and H. Park, "Feature extraction via generalized uncorrelated linear discriminant analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1312-1322, 2006.
[15] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.
[16] J. Shawe-Taylor and S. Sun, "Kernel methods and support vector machines," *Book Chapter for E-Reference Signal Processing*, Elsevier, 2013.
[17] D. R. Hardoon, S. Szedmak and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639-2664, 2004.
[18] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385-2404, 2000.
[19] Z. Liang and P. Shi, "Uncorrelated discriminant vectors using a kernel method," *Pattern Recognition*, vol. 38, no. 2, pp. 307-310, 2005.
[20] T. Sun, S. Chen, Z. Jin and J. Yang, "Kernelized discriminative canonical correlation analysis," *in Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition*, 2007, Beijing, China, pp. 1283-1287.
[21] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," *in Proceedings of the Conference on Data Mining and Data Warehouses*, 2010, Ljubljana, Slovenia, pp. 1-4.
[22] P. Horst, "Relations amongm sets of measures," *Psychometrika*, vol. 26, no.2, pp. 129-149, 1961.
[23] A. Sharma, A. Kumar, H. Daume and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," *in Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2012, Providence, Rhode Island, pp. 2160-2167.
[24] X. Zhang and D. Chu, "Sparse uncorrelated linear discriminant analysis," *in Proceedings of the International Conference on Machine Learning*, 2013, Atlanta, USA, pp. 45-52.
[25] Q. Sun, S. Zeng, Y. Liu, P. Heng and D. Xia, "A new method of feature fusion and its application in image recognition," *Pattern Recognition*, vol. 38, no. 12, pp. 2437-2448, 2005.
[26] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165-175, 1989.
[27] B. Scholkopf, A. Smola and K. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.
[28] D. Cai, X. He, Y. Hu, J. Han and T. Huang, "Learning a spatially smooth subspace for face recognition," *in Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007, Minneapolis, Minnesota, USA, pp. 1-7.
[29] D. Cai, X. He, Y. Hu and J. Han, "Spectral regression: A unified approach for sparse subspace learning," *in Proceedings of the International Conference of Data Mining*, 2007, Omaha NE, USA, pp. 73-82.
[30] C. Liu, W. Hsaio, C. Lee and F. Gou, "Semi-supervised linear discriminant clustering," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, pp. 989-1000, 2014.
[31] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1-30, 2006.

Shiliang Sun is a professor at the Department of Computer Science and Technology and the head of the Pattern Recognition and Machine Learning Research Group, East China Normal University. He received the Ph.D. degree in pattern recognition and intelligent systems from Tsinghua University, Beijing, China, in 2007. He is a member of the PASCAL (Pattern Analysis, Statistical Modelling and Computational Learning) network of excellence, and on the editorial boards of multiple international journals including Neurocomputing and IEEE Transactions on Intelligent Transportation Systems.

His research interests include kernel methods, learning theory, multi-view learning, approximate inference, sequential modeling and their applications, etc

Xijiong Xie is a Ph.D. student in the Pattern Recognition and Machine Learning Research Group, Department of Computer Science and Technology, East China Normal University.

His research interests include kernel methods, support vector machines, etc.



Mo Yang is a master student in the Pattern Recognition and Machine Learning Research Group, Department of Computer Science and Technology, East China Normal University.

Her research interests include multi-view learning, feature extraction, etc.