Modified Principal Component Analysis: An Integration of Multiple Similarity Subspace Models

Zizhu Fan, Yong Xu, Wangmeng Zuo, Jian Yang, Jinhui Tang, Zhihui Lai, and David Zhang, *Fellow, IEEE*

***Abstract*— We modify the conventional principal component analysis (PCA) and propose a novel subspace learning framework, modified PCA (MPCA), using multiple similarity measurements. MPCA computes three similarity matrices exploiting the similarity measurements: 1) mutual information; 2) angle information; and 3) Gaussian kernel similarity. We employ the eigenvectors of similarity matrices to produce new subspaces, referred to as similarity subspaces. A new integrated similarity subspace is then generated using a novel feature selection approach. This approach needs to construct a kind of vector set, termed weak machine cell (WMC), which contains an appropriate number of the eigenvectors spanning the similarity subspaces. Combining the wrapper method and the forward selection scheme, MPCA selects a WMC at a time that has a powerful discriminative capability to classify samples. MPCA is very suitable for the application scenarios in which the number of the training samples is less than the data dimensionality. MPCA outperforms the other state-of-the-art PCA-based methods in terms of both classification accuracy and clustering result. In addition, MPCA can be applied to face image reconstruction. MPCA can use other types of similarity measurements. Extensive experiments on many popular real-world data sets, such as face databases, show that MPCA achieves desirable classification results, as well as has a powerful capability to represent data.**

***Index Terms*— Feature extraction, modified principal component analysis (MPCA), similarity measurement, similarity subspace, weak machine cell (WMC).**

I. Introduction

OVER the past few decades, subspace learning methods have been widely used in face recognition [1] and other applications. These methods are often closely related to the feature extraction. In general, subspace learning first extracts the features from the samples via one or more feature extractors, e.g., principal component analysis (PCA) [2], linear discriminant analysis (LDA) [3], and so on. Then, a classifier, such as the nearest neighbor (NN) classifier, is used to classify the extracted features. Typical subspace learning methods include PCA, LDA, local preserving projection (LPP) [4], and so on.

PCA tries to find a subspace in which the variance is maximized [4]. The PCA subspace is spanned by the eigenvectors corresponding to the leading eigenvalues of the sample covariance matrix. PCA can be applied to both supervised and unsupervised learning. It has been used with success in numerous applications and research areas. Since Turk and Pentland [2] applied PCA (i.e., the conventional PCA) to face recognition, various improvements to PCA have been proposed to enhance its performance or efficiency. In the conventional PCA (hereafter it is referred to as PCA for simplicity), the images need to be converted to the vectors. This scheme may destroy the underlying spatial information within the images. To address this problem, Yang *et al.* [5] proposed a 2-D PCA (2-DPCA) in which the images need not to be converted to the vectors. 2DPCA is usually more efficient than PCA, and can be viewed as a special case of the multilinear PCA [6]. Recently, researchers proposed L1_norm-based PCA, such as PCA_L1 [7], sparse PCA [8], robust PCA [9], and 2DPCA_L1 (2DPCA based on L1_norm) [10], which are robust to outliers. Most of the above PCA approaches may not lead to desirable classification results when they deal with the real-world nonlinear data. As the nonlinear PCA [1], [11], kernel PCA (KPCA) [12], and their variants can effectively capture the nonlinear information, they may provide more powerful ability to deal with the real-world nonlinear data.

Manuscript received November 10, 2012; revised September 30, 2013; accepted December 1, 2013. Date of publication January 2, 2014; date of current version July 14, 2014. This work was supported in part by the 973 Program under Grant 2014CB347600, in part by the National Science Fund for Distinguished Young Scholars under Grant 61125305 and Grant 61233011, in part by the National Science Foundation of China under Grant 61263032, Grant 61071179, Grant 71262011, Grant 61020106004, Grant 60902099, Grant 61362031, Grand 61332011, and Grand 61203376, and in part by the Shenzhen Municipal Science and Technology Innovation Council under Grant JC 201005260122A, and Grant JCYJ20120613153352732 as well as the Jiangxi Provincial Science and Technology Foundation of China under Grant KJLD12067.

Z. Fan is with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the School of Basic Science, East China Jiaotong University, Nanchang 330013, China (e-mail: zzfan3@163.com).

Y. Xu and Z. Lai are with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China, and also with the Key Laboratory of Network Oriented Intelligent Computation, Shenzhen 518055, China (e-mail: yongxu@ymail.com; lai_zhi_hui@163.com).

W. Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com).

J. Yang and J. Tang are with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 200094, China (e-mail: csjyang@mail.njust.edu.cn; jinhuitang@mail.njust.edu.cn).

D. Zhang is with the Biometrics Research Centre, Department of Computing, Hong Kong Polytechnic University, Hong Kong (e-mail: csdzhang@comp.polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2294492

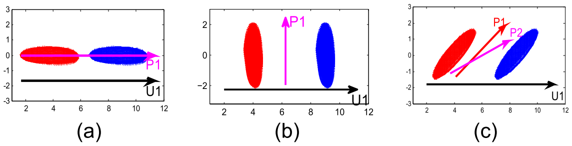

Fig. 1. Three types of relationship between the representative and the discriminative vectors. (a) Representative vector is parallel to the discriminative vector. (b) Representative vector is perpendicular to the discriminative vector. (c) Angle between the representative and the discriminative vectors is a sharp angle.

Independent component analysis (ICA) [13], probabilistic PCA (PPCA) [14], half-quadratic PCA [15], and bilinear PPCA [16] based on the probabilistic similarity measurement are also viewed as the extensions of PCA.

The goal of LDA is to use a criterion, such as Fisher criterion, to determine a number of discriminative vectors. LDA exploits discriminative vectors as transformation axes to transform samples into a new space, i.e., the LDA subspace. These vectors maximize the ratio of the between-class distance to within-class distance in the new space. In 1936, Fisher [3] first presented the basic idea of LDA. LDA has been one of the most important learning methods since about 1990s. Numerous LDA-based approaches and their variants [17]–[20] had been proposed to resolve different problems encountered in face recognition and other classification tasks after [21] proposed Fisherfaces. For example, the approach proposed by Ref. [17] addressed the well-known small sample size problem usually encountered in face recognition. In addition, a local LDA framework we proposed in [20] deals well with the high-dimensional and large-scale data sets. To capture the nonlinear information within the real-world data, Mika *et al.* [22] presented the kernel Fisher discriminant analysis. Similar to 2DPCA, 2-D LDA presented in [23] is able to preserve the spatial structure of data [24]. In addition to the PCA and LDA subspace learning methods, there are also other subspace learning methods, such as LPP subspace [4], [25], probabilistic subspace [26], and random subspace [27], [28].

Although both PCA and LDA are two major methods of feature extraction and dimensionality reduction, the goals of their solutions are different in general. It is well known that PCA aims to find the most representative vectors, i.e., the eigenvectors corresponding to the top eigenvalues of the sample covariance matrix, whereas LDA tries to seek the optimal discriminative vectors of the data. In general, when researchers investigate the PCA-based methods, they usually focus on the most representative vectors nearly ignoring the discriminative vectors. On the other hand, when researchers study the LDA-based methods, they often pay more attention on extracting the optimal discriminative vectors. Actually, there exists some relationship between the representative and the discriminative vectors. To illustrate this relationship, Fig. 1 shows a simple example of a 2-D two-class Gaussian data distribution. In Fig. 1, the red ellipse represents the first class and the blue one represents the second class. P1 is the most representative vector obtained by PCA. U1 is a vector parallel to the optimal discriminative vector obtained by LDA and both lie in the same direction. Obviously, U1 plays the same roles as the optimal discriminative vector in the classification. We consider three types of relationship between the representative and the discriminative vectors. Fig. 1(a) shows the first type of relationship in which the representative vector is parallel to the discriminative vector, and their directions are identical. Hence, if we employ the representative vector to classify the samples, the classification result is the same as that obtained using the discriminative vector. In this sense, representative vectors are equivalent to discriminative vectors and they have the same discriminative capability to classify samples. In other words, the representative vectors contain as much discriminative information as the discriminative vectors do.

In contrast, as shown in Fig. 1(b), the second type of relationship is that the representative vector is perpendicular to the discriminative vector. The classification results obtained by employing the representative vector are far worse than those obtained using the discriminative vector, because the points projected onto the discriminative vector can be easily separated into two classes whereas those projected onto the representative vector are overlapped. In this case, we can conclude that the representative vectors contain a little discriminative information and using them can hardly classify the samples correctly. Fig. 1(c) shows the third type of relationship, which is a more general case. There exists a sharp angle between the representative and the discriminative vectors. We can conclude that if we exploit the representative vectors in Fig. 1(c), the classification results are better than those obtained using the representative vectors in Fig. 1(b) but worse than the classification results obtained using P1 in Fig. 1(a). Similarly, in Fig. 1(c), if we use the representative vector P2 obtained by some other PCA method, e.g., PCA_L1 [7], the classification results are better than those obtained using P1 in principle. Therefore, if we obtain the appropriate representative vectors like P2, which contain more discriminative information than P1 does in Fig. 1(c), we can improve the classification results of PCA without significantly degrading the capability to represent the data.

This paper focuses mainly on how to improve the classification performance of PCA. By borrowing the idea of the graph embedding methods [29], we modify the conventional PCA subspace learning and propose a novel PCA-based subspace learning framework using several similarity measurements. Our framework can achieve much better classification results as well as have a powerful ability to represent the data. We know that each of above mentioned PCA-based subspace methods is based on only one measurement, i.e., the distance or probability measurement. Although it is easy to implement these PCA-based subspace algorithms, the classification results of them are usually not very good. These subspaces may not effectively capture the difference between the samples from different classes. In other words, they may not contain sufficient discriminative information to separate the samples.

We know that the discriminative information contained in one subspace is tightly related to the transformation axes within this subspace. For a classification algorithm, different subspaces may contain different types of the discriminative information. To obtain more discriminative information, we exploit multiple similarity measurements and compute the similarity matrices, which are often used in the graph embedding approaches. We use the eigenvectors of the similarity matrices to produce new subspaces, referred to as similarity subspaces. Each similarity matrix is associated with one kind of similarity measurements. The representative vectors of a similarity subspace (i.e., the vectors spanning this similarity subspace) might contain some discriminative information. In theory, the representative vectors of multiple subspaces contain more discriminative information than those of each of these subspaces do [30]. Therefore, if we effectively capture the discriminative information existed in the multiple PCA-based similarity subspaces, we can obtain desirable classification results.

Our framework produces PCA-based subspace using three similarity measurements: 1) the mutual information; 2) angle information (i.e., cosine distance); and 3) Gaussian kernel distance measurements. These subspaces contain three types of information: 1) the mutual information measuring the information that one variable or sample contains about another one [31], and accounting for the higher order statistics [32]; 2) the cosine distance reflecting the correlation between the samples; and 3) the Gaussian kernel distance capturing the nonlinear information within the data. After producing multiple PCA-based subspaces with the three measurements, we need to resolve the problem of how to effectively capture the discriminative information existed in these subspaces. In PCA, 2DPCA, and KPCA, we often select the representative vectors corresponding to the top eigenvalues to capture the discriminative information and classify the samples. However, it is not always that the representative vectors corresponding to the larger eigenvalues yield the better classification results (for details, please see Section III-C). In essence, the discriminative information of a representative vector is directly related to the classification result yielded by this vector. The high classification accuracy yielded by a representative vector implies that the vector contains much discriminative information, and vice versa. Hence, we can select the representative vectors based on the classification results yielded by these vectors rather than their associated eigenvalues.

Motivated by the weak learning theory [33], we propose a novel feature selection approach to select the representative vectors of the three similarity subspaces. We choose a number of representative vectors potentially yielding high classification accuracies to construct a representative vector set and refer to this set as weak machine cell (WMC). In general, WMC contains more discriminative information than the single representative vector does. Combining the wrapper method and the forward selection scheme, we select one WMC at a time. We integrate the three proposed subspaces into a new similarity subspace through selecting the WMCs that may contain the most discriminative information of the total representative vectors. The classification result yielded by this integrated similarity subspace would be much better than that yielded by the other PCA-based subspace learning methods.

MPCA uses multiple similarity measurements to generate multiple subspaces. It is essentially a similarity subspace learning framework. MPCA has the following ideal properties. First, unlike the other PCA-based subspace methods that use only one measurement, MPCA indeed exploits multiple measurements and provides the strong ability to capture sufficient discriminative information within the data. In addition, it has a powerful capability to represent the data. Second, MPCA is very suitable for the application scenarios in which the number of the training samples is smaller than the dimensionality of the data. Third, similar to the other PCA-based methods, MPCA can theoretically be applied to both supervised and unsupervised learning scenarios. In addition, the similarity measurements in MPCA can have also other forms, e.g., Mahalanobis distance [34]. In short, MPCA provides an in-depth understanding of the PCA-based methods and a new way for modifying the traditional PCA method, from the viewpoint of graph embedding learning. Extensive experiments conducted on many popular real-world data sets show that our framework outperforms the other PCA-based subspace methods in terms of both classification accuracy and clustering result, and can achieve similar or better performance in comparison with the state-of-the-art discriminant analysis algorithms, such as LDA.

The rest of this paper is organized as follows. Section II reviews the PCA. Section III presents three novel similarity subspace models and the similarity subspace learning framework. Section IV presents the details of the experimental results. Section V offers our conclusion.

II. Review of the PCA

Since Pearson [35] first invented and defined PCA through approximating multivariate distributions by lines and planes in 1901, researchers have defined PCA from different aspects [36], [37]. Among these definitions, using covariance matrix of the training samples to define PCA is very popular in pattern recognition and machine learning community. Next, we introduce this definition of PCA. Suppose that there are N centered training samples $x_i \in R^M (i = 1, 2, \ldots, N)$. The covariance matrix of the training set is defined by

$$C = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T = \frac{1}{N} X X^T \tag{1}$$

where $X = [x_1, x_2, \ldots, x_N]$. If the dimensionality of the covariance matrix C is so high (usually $M \gg N$) that the eigen decomposition of C is very difficult or even infeasible, we need to define a new matrix

$$D = \frac{X^T X}{N}. \tag{2}$$

It is easy to prove that two matrices C and D have the same nonzero eigenvalues denoted by $\lambda_i (i = 1, 2, \ldots, r)$. We denote the eigenvectors associated with the nonzero eigenvalues of the matrix D by $v_i (i = 1, 2, \ldots, r)$. Thus, the eigenvectors corresponding to the nonzero eigenvalues of the covariance matrix C are

$$u_i = \frac{X v_i}{\sqrt{\lambda_i}}, \qquad (i = 1, 2, \ldots, r). \tag{3}$$

If PCA is applied to face recognition, then $u_i (i = 1, 2, \ldots, r)$ are called eigenfaces [2] and the subspace spanned by $u_i (i = 1, 2, \ldots, r)$ is usually called eigenspace. Although PCA can be implemented by the more efficient method, i.e., the singular value decomposition [38], the manner of using covariance and correlation matrices can be easily modified to our similarity subspace learning framework, as described in the following section.

III. Similarity Subspace Learning Framework

In this section, we first introduce the basic idea of our MPCA framework. Second, we present three subspace models that correspond to three similarity measurements, respectively. Third, using a novel feature selection algorithm, we integrate these three subspace models into a new PCA-based similarity subspace framework, i.e., the MPCA framework. Finally, we give the implementation of our MPCA framework.

A. Basic Idea of Similarity Subspace Framework

Recently, the graph embedding learning methods have attracted much attention in machine learning community [29]. In graph embedding, the data set is expressed as a graph with each vertex denoting a data point or sample. Graph embedding uses one similarity measure to compute a similarity matrix that is applied to build the learning model. In [29] and [39], the classical PCA is reformulated as one type of the graph embedding model. From the definition of the matrix D in Section II, we know that each entry of the D is the correlation of pairwise data points. Hence, D can be called the correlation matrix. According to (3), the eigenvectors of the covariance matrix can be derived from those of the correlation matrix. Indeed, the correlation is one type of the similarity measurement. As a result, the correlation matrix D is a case of the similarity matrices. Therefore, the model using (2) and (3) is also one type of the graph embedding learning approach.

For graph embedding, we can exploit various similarity measures to yield different similarity matrices [29]. These matrices may generate different types of discriminative and representative information, since they can yield different transformation axes. If we select some appropriate similarity measurements, we may obtain a number of similarity matrices that generate more discriminative information than the correlation matrix D, and have a powerful ability to represent the data. Therefore, the correlation matrix D can be replaced by other types of similarity matrix S in many real-world applications. When the matrix D is replaced by the matrix S, the following problem will arise: whether there exists a matrix W, referred to as similarity driven matrix (SDM), which has the same nonzero eigenvalues as the matrix S. If the matrix W exists, we can use the eigenvectors of the similarity matrix S to construct a novel subspace and exploit this subspace to classify or represent the data. This is similar to the traditional PCA that performs in the subspace spanned by the eigenvectors of the covariance matrix C. Fortunately, the theorem we propose, i.e., Theorem 1, guarantees that there usually exist the Ws. Thus, for one type of similarity matrix, we can obtain its corresponding PCA-based learning model.

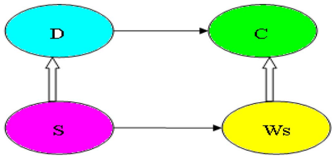

Fig. 2. Relationship between the matrices *D, C, S,* and Ws.

Fig. 2 shows the relationship between the matrices D, C, S, and Ws. Among these matrices, the matrices C and D have the same nonzero eigenvalues. The eigenvectors associated with the nonzero eigenvalues of the matrix C are derived from those of the matrix D. The matrix S is a similarity matrix and the matrix D is a case of the matrix S. Similarly, the relationship between the matrix S and the matrices Ws is the same as that between the matrices D and C.

In Fig. 2, if we want to compute the eigenvectors corresponding to the nonzero eigenvalues of the covariance matrix C that is very large scale when the samples are very high dimensional, we can first compute the eigenvectors associated with the nonzero eigenvalues of the correlation matrix D. Then, we determine the eigenvectors corresponding to the nonzero eigenvalues of the matrix C. Similarly, the matrix C may be viewed as a special instance of the matrices Ws that are obtained by the eigenvalues and eigenvectors of the matrix S. The following theorem describes the relationship between the similarity matrix S and its corresponding matrices Ws.

Theorem 1: Let $X = [x_1, x_2, \ldots, x_N]$ where $x_i \in R^M (M > N)$ denote the centered training set. The nonzero eigenvalues and their eigenvectors of the similarity matrix S of size $N \times N$ are λ_i and $\alpha_i (i = 1, 2, \ldots, r)$, respectively. Let $\beta_i = X\alpha_i/\sqrt{\lambda_i}, (i = 1, 2, \ldots, r)$. If these vectors are linearly independent, then there exists one or more matrices Ws of size $M \times M$ that have eigenvectors β_i corresponding to $\lambda_i (i = 1, 2, \ldots, r)$, and the Ws are determined by the data set X.

Proof: Given an arbitrary matrix W of size $M \times M$, if its nonzero eigenvalues are $\lambda_i (i = 1, 2, \ldots, r)$, and its eigenvectors corresponding to these nonzero eigenvalues are $\beta_i (i = 1, 2, \ldots, r)$, then we can establish the following equation systems:

$$W\beta_i = \lambda_i \beta_i, \quad (i = 1, 2, \ldots, r) \tag{4}$$

where

$$W = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1M} \\ w_{21} & w_{22} & \cdots & w_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & \cdots & w_{MM} \end{bmatrix}, \quad \beta_i = \begin{pmatrix} b_{i1} \\ b_{i2} \\ \vdots \\ b_{iM} \end{pmatrix}.$$

Then, we can obtain M equation systems, and the jth $(j = 1, 2, \ldots, M)$ one is as follows:

$$\mathrm{BW}_j = Y_j, \quad (j = 1, 2, \ldots, M) \tag{5}$$

where $Y_j = (\lambda_1 b_{1j}, \lambda_2 b_{2j}, \ldots, \lambda_r b_{rj})^T$

$$B = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_r \end{pmatrix} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1} & b_{r2} & \cdots & b_{rM} \end{pmatrix} \quad W_j = \begin{pmatrix} w_{j1} \\ w_{j2} \\ \vdots \\ w_{jM} \end{pmatrix}.$$

According to the assumption, $\beta_i \in R^M (i = 1, 2, \ldots, r)$ are linearly independent, and $\text{rank}(B) = \text{rank}(BY_j) < N < M$. This guarantees that the linear system (5) has infinitely many solutions. Therefore, it is certain that the Ws satisfying the conditions in Theorem 1 exist. According to (4) and (5), the Ws are determined by λ_i and $\beta_i (i = 1, 2, \ldots, r)$, and $\beta_i = X\alpha_i/\sqrt{\lambda_i}$. In addition, α_i is obtained by X. Thus, the Ws are determined by X. ■

Remark 1: Theorem 1 guarantees the matrices Ws existing when $\beta_i (i = 1, 2, \ldots, r)$ are linearly independent. According to the theory of the linear algebra, if the number of samples is greater than their dimensionality, then these samples are linearly dependent. Otherwise, these samples are very likely to be linearly independent, particularly when their dimensionality is far higher than the number of the samples. Notice that in the real-world applications, such as face recognition, the training procedure usually uses a small portion of the data set. As a consequence, the dimensionality of the samples is generally far higher than the number of the samples, as supposed in Theorem 1. Therefore, it is reasonable to suppose that the $\beta_i (i = 1, 2, \ldots, r)$ are linearly independent in the high-dimensional real-world applications. In this case, matrices Ws always exist. Thus, we can use the similarity subspaces corresponding to the Ws and perform the feature extraction and classification.

Like the kernel-based methods in which the nonlinear mapping is not explicit and needs to be specified by some kernel function, one instance of the Ws can be obtained by some implicit computation on the data set X. This instance is explicitly expressed by its corresponding similarity matrix. In some cases, the Ws can also be explicitly determined. For example, the covariance matrix C in PCA can be viewed as one instance of the Ws and it can be explicitly computed using (1).

In general, if one type of the similarity measurements is given, we can therefore determine one type of the SDMs. As described in Section I, we employ three similarity measurements to compute three types of the similarity matrices. Using Theorem 1, we can determine three types of the SDMs. The eigenvectors of each type of SDMs span a similarity subspace. Thus, we obtain three similarity subspaces.

We believe that the discriminative capability of the combination of the three similarity subspaces is not weaker than that of one of three similarity subspaces. Thus, if we properly integrate these three subspaces into a new similarity subspace, the discriminative capability of this new subspace might also not be weaker than that of one of three similarity subspaces. In practice, we observe that the new integrated subspace is much stronger than one of three similarity subspaces and other PCA-based subspaces in terms of the discriminative capability.

B. Similarity Subspace Models

1) Similarity Subspace Model Based on Mutual Information: In information theory, entropy and mutual information are two basic concepts. We first give some definitions about them [40], [41]. Here, we only consider the discrete situation.

Definition 1 (Entropy): Given a discrete random variable x, the entropy $H(x)$ is defined by

$$H(x) = -\sum_{x\in\chi} p(x)\log p(x). \tag{6}$$

Here, $p(x)$ is the probability density function of random variable x.

Definition 2 (Joint Entropy): Given a pair of discrete random variables (x, y) with a joint distribution $p(x, y)$, the joint entropy $H(x, y)$ is defined by

$$H(x, y) = -\sum_{x\in\chi}\sum_{y\in\gamma} p(x, y)\log p(x, y). \tag{7}$$

Definition 3 (*Mutual Information*): Given two random variables x and y with a joint distribution $p(x, y)$, their marginal probability functions are $p(x)$ and $p(y)$, respectively. The mutual information $I(x, y)$ is defined by

$$I(x, y) = -\sum_{x\in\chi}\sum_{y\in\gamma} p(x, y)\log\left(\frac{p(x, y)}{p(x)p(y)}\right). \tag{8}$$

In this paper, we use the normalized mutual information [42], which is defined by

$$\text{NI}(x, y) = \frac{I(x, y)}{\min(H(x), H(y))}. \tag{9}$$

We use (9) to measure the similarity between two training samples. Thus, the similarity matrix is defined by

$$\text{SM} = (\text{NI}(x_i, x_j)), \quad (i, j = 1, 2, \ldots, N). \tag{10}$$

The nonzero eigenvalues and their eigenvectors of the similarity matrix SM are λ_j^m and $\alpha_j^m (j = 1, 2, \ldots, q)$, respectively. Then, the similarity subspace based on mutual information is spanned by the following vectors:

$$\beta_j^m = \frac{X\alpha_j^m}{\sqrt{\lambda_j^m}}, \quad (j = 1, 2, \ldots, q) \tag{11}$$

where the data set X is vector based. That is, the data set is expressed as the vectors. This first similarity subspace model is referred to as MPCA model 1 (MPCA1).

2) Similarity Subspace Model Based on Cosine Distance: In the second similarity subspace model, we use cosine distance to measure the similarity between the data points. The similarity matrix is defined by

$$\text{SC} = (\cos(x_i, x_j)), \quad (i, j = 1, 2, \ldots, N) \tag{12}$$

where $\cos(x_i, x_j)$ denotes the cosine distance between two data points x_i and x_j, i.e., $\cos(x_i, x_j) = x_i^T x_j/(\|x_i\| \cdot \|x_j\|)$. We assume that the nonzero eigenvalues and their eigenvectors of the similarity matrix SC are λ_l^c and $\alpha_l^c (l = 1, 2, \ldots, s)$, respectively. Then, the similarity subspace based on cosine distance is spanned by the following vectors:

$$\beta_l^c = \frac{X\alpha_l^c}{\sqrt{\lambda_l^c}}, \quad (l = 1, 2, \ldots, s) \tag{13}$$

where the data set X is also vector based. This similarity subspace model is referred to as MPCA model 2 (MPCA2). Note that if the sample vectors are normalized, or they have unit L2 norm, the MPCA2 reduces to PCA. In addition, PCA can be viewed as a special case of MPCA2.

3) Similarity Subspace Model Based on Kernel Distance: With the similar idea, we use the Gaussian kernel distance, i.e., Gaussian kernel function, to measure the similarity between the data points in the third model. The similarity matrix is defined by

$$\mathrm{SK} = (k(x_i, x_j)), \quad (i, j = 1, 2, \ldots, N) \tag{14}$$

where $k(x_i, x_j)$ denotes the Gaussian kernel distance between two data points x_i and x_j, i.e., $k(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$, where σ is the kernel parameter needed to be specified in practice. Similarly, we assume that the nonzero eigenvalues and their eigenvectors of the similarity matrix SK are λ_h^k and $\alpha_h^k (h = 1, 2, \ldots, t)$, respectively. Then, the similarity subspace based on the kernel distance is spanned by the following vectors:

$$\beta_h^k = \frac{X\alpha_h^k}{\sqrt{\lambda_h^k}}, \quad (h = 1, 2, \ldots, t) \tag{15}$$

where the data set X is also vector based. This similarity subspace model is referred to as MPCA model 3 (MPCA3).

Among three similarity measures, the mutual information accounts for the higher order statistics, which can be applied to the nonlinear transformations [32], whereas the correlation used in the traditional PCA accounts for the second-order statistics, which is suitable for learning the linear separable data [31]. The proposed MPCA1 model using the mutual information is more suitable for dealing with the nonlinear data sets than the traditional PCA in principle. The cosine distance between samples is an extension of the correlation between samples. Compared with the traditional PCA, one of our proposed models, i.e., MPCA2, which uses the cosine distance between samples as the similarity measure, may be suitable for more applications. We know that the Gaussian kernel distance is nonlinear and can capture the nonlinear information within the data set. This similarity measure is very suitable for dealing with the nonlinear data sets. The proposed three similarity subspaces can produce three types of transformation axes. They may contain complementary discriminative information to some extent. For the above reasons, we select these three similarity measurements. From the point of view of graph embedding learning, the above MPCA models correspond to three graph embedding learning models, respectively.

Next, we will investigate the discriminative information contained in the above three similarity subspaces and the eigenspace in the traditional PCA. The discriminative information contained in a subspace is tightly related to the classification accuracy yielded by this subspace in a classification algorithm (e.g., the NN classifier). If one subspace can achieve higher classification accuracy, then it contains more discriminative information, and vice versa. Given an eigenspace E, suppose there exist the vectors $e_1, e_2, \ldots, e_l$ in E yielding the highest classification accuracy $D(e_1, e_2, \ldots, e_l)$ in a classification algorithm. We define $D(e_1, e_2, \ldots, e_l)$ as the discriminative information contained in the space E, which is denoted by $D_E (D_E = D(e_1, e_2, \ldots, e_l))$. Similarly, let D_S be the discriminative information contained in the similarity subspace spanned by (11), (13), and (15). That is, D_S is the highest classification accuracy yielded by the vectors in the above similarity subspaces in a classification algorithm. Thus, we have the following theorem.

Theorem 2: If three similarity matrices SM, SC, and SK have more than N linearly independent eigenvectors, then $D_S \geq D_E$, in a classification algorithm.

Proof: Suppose that three similarity matrices *SM*, *SC*, and *SK* have more than N linearly independent eigenvectors, and N eigenvectors denoted by $\xi_1, \xi_2, \ldots, \xi_N$ are chosen. Hence, according to the basic linear algebra theory, each eigenvector of D defined by (2), $\alpha_i (i = 1, 2, \ldots, r)$, can be linearly represented by $\xi_1, \xi_2, \ldots, \xi_N$. That is

$$\alpha_i = p_1\xi_1 + p_2\xi_2 + \cdots + p_N\xi_N$$

where $p_i \in R (i = 1, 2, \ldots, N)$. Since $\xi_1, \xi_2, \ldots, \xi_N$ are a part of the eigenvectors of three similarity matrices *SM*, *SC*, and *SK*, denoted by $A = (\alpha_1^m, \ldots, \alpha_q^m, \alpha_1^c, \ldots, \alpha_s^c, \alpha_1^k, \ldots, \alpha_t^k)$, each α_i can be linearly represented by A. That is

$$\begin{aligned}\alpha_i &= z_{m1}\alpha_1^m + \cdots + z_{mq}\alpha_q^m + z_{c1}\alpha_1^c + \cdots + z_{cs}\alpha_s^c \\ &\quad + z_{k1}\alpha_1^k + \cdots + z_{kt}\alpha_t^k = \sum_{j=1}^{q} z_{mj}\alpha_j^m + \sum_{l=1}^{s} z_{cl}\alpha_l^c + \sum_{h=1}^{t} z_{kh}\alpha_h^k.\end{aligned}$$

According to Theorem 1

$$\begin{aligned}\beta_i &= X\alpha_i/\sqrt{\lambda_i} \\ &= \sum_{j=1}^{q} z_{mj} X \frac{\alpha_j^m}{\sqrt{\lambda_i}} + \sum_{l=1}^{s} z_{cl} X \frac{\alpha_l^c}{\sqrt{\lambda_i}} + \sum_{h=1}^{t} z_{kh} X \frac{\alpha_h^k}{\sqrt{\lambda_i}} \\ &= \sum_{j=1}^{q} z_{mj} X \frac{\alpha_j^m}{\sqrt{\lambda_j^m}} \cdot \frac{\sqrt{\lambda_j^m}}{\sqrt{\lambda_i}} + \sum_{l=1}^{s} z_{cl} X \frac{\alpha_l^c}{\sqrt{\lambda_l^c}} \cdot \frac{\sqrt{\lambda_l^c}}{\sqrt{\lambda_i}} \\ &\quad + \sum_{h=1}^{t} z_{kh} X \frac{\alpha_h^k}{\sqrt{\lambda_h^k}} \cdot \frac{\sqrt{\lambda_h^k}}{\sqrt{\lambda_i}} \\ &= \sum_{j=1}^{q} z_{mj} \frac{\sqrt{\lambda_j^m}}{\sqrt{\lambda_i}} \beta_j^m + \sum_{l=1}^{s} z_{cl} \frac{\sqrt{\lambda_l^c}}{\sqrt{\lambda_i}} \beta_l^c + \sum_{h=1}^{t} z_{kh} \frac{\sqrt{\lambda_h^k}}{\sqrt{\lambda_i}} \beta_h^k.\end{aligned}$$

Let

$$\begin{aligned} z'_{mj} &= z_{mj}\sqrt{\lambda_j^m}/\sqrt{\lambda_i} \in R, \quad (j = 1, 2, \ldots, q) \\ z'_{cl} &= z_{cl}\sqrt{\lambda_l^c}/\sqrt{\lambda_i} \in R, \quad (l = 1, 2, \ldots, s)\end{aligned}$$

and

$$z'_{kh} = z_{kh}\sqrt{\lambda_h^k}\Big/\sqrt{\lambda_i} \in R, \quad (h = 1, 2, \ldots, t).$$

Then, we have

$$\beta_i = \sum_{j=1}^{q} z'_{mj}\beta_j^m + \sum_{l=1}^{s} z'_{cl}\beta_l^c + \sum_{h=1}^{t} z'_{kh}\beta_h^k.$$

Therefore, we can conclude that each eigenvector of C in (1) can be linearly represented by

$$B = (\beta_1^m, \ldots, \beta_q^m, \beta_1^c, \ldots, \beta_s^c, \beta_1^k, \ldots, \beta_t^k).$$

Hence, the eigenspace is a subset of the similarity subspace spanned by B. Then, the vectors $e_1, e_2, \ldots, e_l$ can be found in this similarity subspace. In a classification algorithm, we can always choose the vector set containing $e_1, e_2, \ldots, e_l$ that yields the classification accuracy D_v such that $D_v \geq D_E$. According to the definition of D_S, we have $D_S \geq D_v$. Thus, $D_S \geq D_E$. ■

Although D_S may include features that are not only valid but also harmful for discrimination, using the proposed WMC can select valid features among them. Thus, D_S can achieve a higher performance than D_E. Theorem 2 justifies that three similarity subspaces contain more (at least not less than) discriminative information than the PCA subspace. In other words, the proposed three subspaces can lead to higher classification accuracy than the PCA subspace in a classification algorithm. In theory, if we orthogonalize the bases in three similarity subspaces, then we can obtain a new subspace that is equivalent to the above three subspaces. That is, each vector in the above three subspaces can be obtained in the new subspace, and vice versa. Therefore, the classification accuracy yielded by the new subspace is the same as that yielded by the three similarity subspaces in a classification algorithm. From the new subspace, we can select an appropriate feature set, which contains the most discriminative information using WMC, and obtain the integrated MPCA subspace. Thus, this integrated MPCA subspace contains more discriminative information than the conventional PCA subspace. That is, MPCA can lead to better classification performance than PCA. Next, we will integrate the proposed three similarity subspaces into the new subspace, i.e., MPCA, via feature selection.

C. Subspace Integration Using Feature Selection

In this section, we present the subspace integration via a novel random feature selection algorithm. First, we give the motivation of our feature selection algorithm. Second, we introduce how to construct the WMCs in the proposed novel feature selection. Finally, we give the integration procedure and the implementation of MPCA.

1) Motivation of Feature Selection Scheme in MPCA: After obtaining the three similarity subspaces, we need to integrate these subspaces into a new similarity subspace via feature selection [43]. In the feature selection, we choose those representative vectors that can effectively capture the difference between the samples from different classes (in other words, they contain sufficient discriminative information) from three MPCA models. There are a great number of feature selection algorithms in the literature that can be roughly grouped into three categories: 1) filters [42]; 2) wrappers [44]; and 3) hybrid algorithms [45]. The filters usually select the features without needing the classifier performance, whereas the wrappers need the classifier performance to select the features. The hybrid approaches are usually the combination of the filter and wrapper approaches [45].

Although it is not difficult to implement the filter approaches, the selected features are not directly related to the classification performance. Note that in the PCA-based methods, we use the representative vectors as the features. Actually, it is not always that the features corresponding to the larger eigenvalues yield the better classification results in PCA. Fig. 3 shows an example of this case and shows the classification results yielded by individual features on two popular face data sets: the AR and the ORL data sets [46], [47]. In Fig. 3, the classification accuracy tends to be low when the eigenvalues are small (the eigenvalues are descending). However, this case does not always hold. For example, in Fig. 3(b), the classification accuracy yielded by the feature associated with the 10th eigenvalue is only 1.5%, whereas the classification accuracy yielded by the one corresponding to the 11th eigenvalue is 5.5%. There are many such cases in Fig. 3. Moreover, since our similarity subspace learning framework involves three similarity subspaces, it is impossible to select the features directly using the eigenvalues. On the other hand, the wrapper approach is directly related to the classification results. MPCA uses this approach to determine those features that yield the high classification accuracy, i.e., they contain relatively much discriminative information for classification.

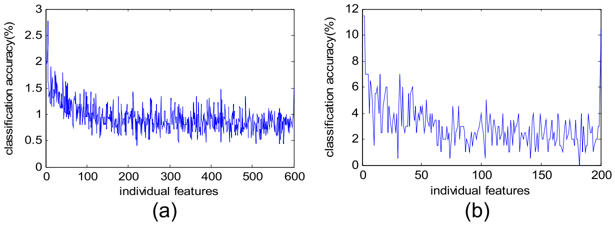

Fig. 3. Classification results yielded by individual features in PCA. (a) AR face data set has 120 individuals. For each individual, the first five samples are used for training and the rest are used for testing. (b) ORL face data set has 40 individuals. For each individual, the first five samples are used for training and the rest are used for testing.

The feature searching technique is the sequential forward selection [48]. We select a number of features at a time (many other wrapper algorithms usually select only one feature at a time, e.g., the algorithm in [44]). In the selection process, we first orthogonalize all the features of three similarity subspaces. From the point of view of probability theory, the classification accuracy obtained by a feature can be viewed as the probability of the event that each test sample can be correctly classified exploiting this feature [33]. Note that the features are orthogonal and a feature does not affect the performance obtained by the other features. In this sense, we assume that the individual features tend to be independent identically distributed.

According to Fig. 3, the classification accuracy obtained by each feature is low. As shown in this figure, for the ORL data set, the highest accuracy is 11.5%, and for the AR data set, the highest accuracy is only 2.78%. Actually, for most large-scale data sets, such as AR and Carnegie Mellon University (CMU)-PIE [49], the classification accuracies obtained by individual features of most subspace methods are usually low. Indeed, if we select the features one by one in the wrapper approach, the selection efficiency will be very low. In addition, this selection scheme may not guarantee to obtain the desirable classification results. On the other hand, if we select too many features at a time, it is very difficult to obtain the desirable classification results. In this paper, we select the features by constructing the WMC that contains an appropriate number of features.

In general, the classification result yielded by simultaneously using those features that contain relatively much discriminative information (i.e., the classification accuracy

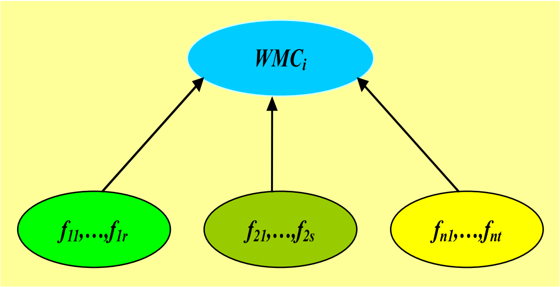

Fig. 4. WMC construction process.

yielded by each of these features is higher than some value, say, 0.5% in this paper) is often better than that yielded using each of these features. The WMC construction is to automatically select such features for each WMC containing sufficient discriminative information, and try to make the discriminative information within each WMC equal. Thus, we can conveniently and automatically determine the final features to perform learning using such WMCs. In the usual random feature selection algorithms, one of the key problems is how to determine an appropriate number of the features n_c at a time. It can be addressed well in our construction of WMC. Our method does not need to know n_c in advance.

2) Construction of WMC: Our proposed feature selection is a random approach. It contains two steps. The first step is to select an appropriate number of features to construct WMC. The second step is to employ a wrapper method to select the constructed WMC.

Fig. 4 shows the construction process of a WMC. For a feature f_i that is a leaf in Fig. 4, if the classification accuracy obtained by f_i on a data set is p_i, we can say that the probability of the event that this feature correctly classifies a given test sample is p_i. Given a feature set F containing n_c features, if it has at least one feature that can correctly classify a test sample, then we consider that F roughly correctly classifies this test sample. Due to the above assumption that the features tend to be independent identically distributed, we can define the probability of the event that F roughly correctly classifies the test sample as

$$P_{\mathrm{wm}} = 1 - \prod_{i=1}^{n_c} (1 - p_i) \tag{16}$$

where P_{wm} is referred to as pseudo recognition rate (PRR) of the set F. Particularly, when all the probabilities p_i are equal, i.e., all of them are p, then

$$P_{\mathrm{wm}} = 1 - (1 - p)^{n_c}. \tag{17}$$

If n_c is fixed, the actual classification result tends to be better when P_{wm} becomes higher.

To construct the WMCs, we set a PRR for each WMC to be some value, which is slightly greater than a threshold T, say, 0.5. This is inspired by the basic idea of the typical boosting algorithms [50], [51]. If the classification accuracy yielded by a feature is greater than T, then we consider this feature is a WMC. Thus, the number of features in this WMC is one. For other features whose corresponding classification accuracies are less than T, we construct the WMCs based on the bottom-up technique, as shown in Fig. 4. First of all, the classification accuracies $\{c_1, \ldots, c_r\}$ yielded by these features are sorted in ascending order. Then, we group these accuracy values into a number of parts. The values of each part belong to some interval. For example, there are 12 accuracy values: 0.01, 0.01, 0.02, 0.05, 0.05, 0.06, 0.1, 0.1, 0.12, 0.15, 0.16, and 0.22. They can be grouped into three parts: 1) (0.01, 0.01, 0.02, 0.05, 0.05); 2) (0.06, 0.1, 0.1, 0.12, 0.15); and 3) (0.16, 0.22). The values of these parts belong to three interval (0, 0.05], (0.05, 0.15], and (0.15, 0.22], respectively.

Algorithm 1 Construction of WMC

```
1.Input: classification accuracies $C = \{c_1, \cdots, c_n\}$
   yielded by the features $f_i$ ($i$ = 1, …, $n$).
2. Initialization: sort the elements in $C$ in ascending
   order. W= {}, $T$ = 0.5.
3. If $c_i \geq T$, then
      $W_i = f_i$, $W = W + \{W_i\}$, $C = C - \{c_i\}$.
4. Group the elements of $C$ into the parts:
      $p_k = (c_{k1}, \cdots, c_{kn_k})$, $(k = 1, \cdots, s)$.
5. For each part $p_k$, specify some value $p$ to substitute
   for all elements in this part.
   1) Use (17) to determine $n_c$.
   2) Repeat: randomly select $n_c$ values ($c'_{k1}, \cdots, c'_{kn_c}$)
   from the part, use their corresponding features to
   construct a WMC, denoted by $W_j$:
         $W = W + \{W_j\}$, $p_k = p_k$ -($c'_{k1}, \cdots, c'_{kn_c}$)
    Till the number of values in this part $|p_k| \leq n_c$.
6. Randomly assign the remaining features in each part
   into the final WMCs, and return the WMC set $W$.
```

From the point of view of the discriminant analysis, we consider that those individual features yielding nearly equal classification accuracies have almost same discriminative information. For simplicity, we view that classification accuracies obtained by these individual features are also same and assume that they are all p_w. If we specify P_{wm} in (17), n_c can be therefore computed as

$$n_c = \left\lceil \log^{(1-P_{\mathrm{wm}})}_{(1-p_w)} \right\rceil$$

where $\lceil x \rceil$ is the smallest integer whose value is larger than x. As discussed above, P_{wm} is set to 0.5. For a part, we randomly select n_c values from it, and use their corresponding features to construct a WMC. Then, we delete these selected n_c values from this part. This procedure repeats till the number of the values in this part is less than n_c. If there are remaining accuracy values in a part, we randomly assign the features corresponding to these values to all of the final WMCs. Algorithm 1 describes the WMC construction procedure.

Algorithm 2 Procedure of WMC Selection

```
1.  Initialization: $W = \{W_i\}$ ( $i = 1,2,..,n_w$ ),
    $S = \varphi$; $\varepsilon$=1e-4; $oa$ = 0, and $na$ = 0;
2.  Randomly select a $W_k$ , $W = W - W_k$ ;
    $S = S + W_k$;
3.  Let $ca(S)$ be the classification accuracy yielded by
    $S$, and $|W|$ denote the number of the WMCs in $W$;
    While ($na$ - $oa \geq 0$ ) and ( $|W| > 1$ )
     $oa = na$;
      for $i$ = 1: $|W|$
         $S_i^{'} = S + W_i$; compute the $ca(S_i^{'})$ ;
      end
        $j = \arg(\max_i ca(S_i^{'}))$ ; $S = S + W_j$ ;
          $W = W - W_j$ ; $na = ca(S)$ - $\varepsilon$ ;
    end
 4. Return $S$;
```

Algorithm 3 MPCA Framework

```
1.  Input the centered training samples:
    $x_i \in R^M$ ( $i = 1,2,...,N$ );
2.  Compute MPCA 1 model using (6)-(11);
3.  Compute MPCA 2 model using (12)-(13);
4.  Compute MPCA 3 model using (14)-(15);
5.  Orthogonalize all the features in the three models ;
6.  Compute the classification accuracy yielded by
    each orthogonalized feature and construct the
    WMCs.
7.  Adopt Algorithm 2 to determine the WMCs
8.  Use the determined WMCs to classify test
    samples.
```

3) Subspace Integration and the MPCA Framework: Before subspace integration, we need to determine the WMCs to classify the test samples. Note that although the PRRs of the WMCs are nearly same, i.e., all of them should slightly greater than the specified threshold T, the discriminative capabilities of the WMCs may be different. Therefore, we need to select those WMCs that have higher discriminative capabilities, i.e., contain more discriminative information. As mentioned earlier, since the wrapper method is directly related to the discriminative capability of the features, we combine this method with sequential forward selection to determine the WMCs.

Algorithm 2 gives the pseudocodes to select the WMCs (assume that the number of the total WMCs is n_W). Since we select WMCs in a random manner, if we randomly select them several times, the selecting results are usually different. We can take the WMCs that yield the highest classification accuracy as the best WMCs. Algorithm 3 describes our MPCA framework.

Note that MPCA is integrated by three PCA-based models: 1) MPCA1; 2) MPCA2; and 3) MPCA3. MPCA is derived from three similarity measures, and is performed in the original input space like the traditional PCA. In concept, KPCA is performed in the high-dimensional feature space usually induced by a nonlinear mapping, which is often defined by one type of the kernel functions. Hence, the feature extractions of these two methods are different.

IV. Experimental Result

In this section, we have conducted the experiments to evaluate the effectiveness of the proposed method. The first two experiments are conducted on two widely used real-world face data sets: 1) the AR and 2) CMU-PIE data sets, respectively. The goal of these experiments is to show the discriminative capability of MPCA. The third experiment is also conducted on these two data sets to show the effectiveness of our feature selection method. The fourth experiment is conducted on the Georgia Tech (GT) face data set to show the unsupervised learning ability of MPCA [47]. The fifth experiment is conducted on the ORL face data set. The goal of this experiment is to show the representative capability of MPCA. Finally, we study the roles of the individual similarity subspaces in classification. In the first three experiments, the classifier we used is the NN classifier based on the Euclidean distance (L2 norm). When constructing the WMCs, we empirically use 0.03 to substitute for each accuracy value in the interval (0, 0.05], 0.1 for the interval (0.05, 0.15], 0.2 for the interval (0.15, 0.25], 0.3 for the interval (0.25, 0.35], and 0.4 for the interval (0.35, 0.5]. The parameters used in the other algorithms are the best ones we obtain via parameter tuning through fivefold cross validation.

For fair comparison, we implemented 19 algorithms in the first two experiments. They are the typical NN classifier in the original space, 1DPCA_L2 (i.e., PCA) [2], 2DPCA_L2 (the 2DPCA based on L2 norm) [5], KPCA [11], 1DPCA_L1 (the PCA based on L1 norm) [7], 2DPCA_L1 [10], ICA [13], LPP [4], LDA [21], [52], kernel LDA (KLDA) [22], [53], subclass discriminant analysis (SDA) [18], feature extraction algorithm based on ICA (ICA-FX) [54], maximum margin criterion (MMC) [55], null LDA (NLDA) [17], Chernoff LDA (CLDA) [56], and our four algorithms: 1) MPCA1; 2) MPCA2; 3) MPCA3; and 4) MPCA framework that integrates three MPCA models. In KPCA, KLDA, and MPCA3, we used the Gaussian kernel function.

A. Experiment on the AR Face Data Set

The first experiment is conducted on a large data set, the AR face data set. It contains over 4000 color face images of 126 individuals [5], [52]. We used the face images of 120 individuals and each individual has 26 images. All the images are cropped with dimension 50 × 40 pixels and converted to gray scale. For each individual, N (= 9, 10, 11, 12, 13) images are randomly selected for training (yielding five training subsets), and the rest are used for testing.

Let *dis* denote the distance between the first training sample of the first individual and the first training sample of the second individual. The kernel function parameters of KPCA and MPCA3 are equal and they are set to 400 * *dis*, and the kernel parameter of KLDA is set to 0.003 * *dis*. In LPP, we need to set the local parameter ε to determine the number of NNs of a sample and it is set to 0.1 * d, where d is the mean of the distance between two arbitrary samples in the training set. Since the features in three similarity subspaces may not be orthogonal, we orthogonalize them when constructing WMCs. This is also conducted in the second experiment.

We randomly ran each algorithm 10 times on each training subset. For the MPCA framework, since the WMC construction is random, to obtain the classification result as optimal as possible, we randomly ran the WMC construction procedure 10 times and selected the best WMCs in terms of the classification accuracy in each run of MPCA. In the WMC selection, we need a supervised learning procedure to choose the final WMCs. We randomly divided each training subset into two equal parts. One is used for training, and the other one is a validation set, which is used to determine the WMCs. We take the training subset $N = 9$ as an example. The number of training samples in this subset is 1080. It is randomly divided into two equal parts. That is, the part for training contains 540 samples, and the validation set also contains 540 samples. In addition, we conducted the same selection procedures in the second experiment.

For each algorithm, Table I reports the best classification results on five training subsets. In this table, the bold italics highlight the best classification result on each training subset. According to Table I, our three models MPCA1, MPCA2, and MPCA3 are comparable with many other PCA-based methods in terms of the classification accuracy. As shown in Table I, the classification results of MPCA are significantly better than those of the other PCA-based algorithms. In addition, MPCA can achieve similar or better performance in comparison with the state-of-the-art supervised algorithms ICA-FX, MMC, NLDA, CLDA, LDA, KLDA, and SDA. That is, from Table I, we can see that when $N = 11$, 12, and 13, the classification results of MPCA are better than those of ICA-FX, MMC, NLDA, CLDA, LDA, KLDA, and SDA. The main reason of good classification performances of MPCA may be that three similarity subspaces defined in our approach contain sufficient discriminative information. Meanwhile, our feature selection based on WMC plays an important role to effectively capture the discriminative information contained in these three subspaces.

B. Experiment on the CMU PIE Face Data Set

The second experiment is conducted on the CMU PIE face data set containing 68 individuals. Each individual has images captured under 13 different poses and 43 different illumination conditions and with four different expressions [49], [57]. We use one near-frontal pose C05 in which all the images are under different illuminations and expressions. The data subset C05 contains 68 individuals and each individual has 49 images. Each image is manually aligned and cropped. The size of each image is 64 × 64 [58]. For each individual, N (= 5, 10, 15, 20) images are randomly selected for training, and the rest are used for testing.

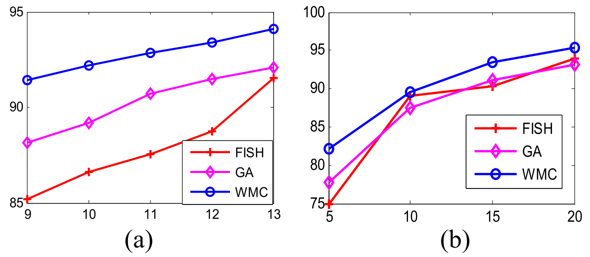

Fig. 5. Classification results of three feature selection approaches. (a) Result on the AR data set. (b) Result on the CMU PIE data set.

We implemented 19 algorithms that are the same as ones used in the first experiment. The kernel function parameters of KPCA and MPCA3 are set to 1000 * *dis* defined in the first experiment, and the kernel parameter of KLDA is set to 0.1 * *dis*. In LPP, the local parameter is set to 0.1 * d that is also defined in the first experiment. Similarly, we randomly ran each algorithm 10 times on each training subset and randomly ran the WMC construction procedure 10 times in each run of MPCA. For each algorithm, Table II reports the best classification results on four training subsets. Like Table I, the bold italics highlight the best classification result on each training subset. From Table II, we find that MPCA significantly outperforms the other PCA-based algorithms. In addition, MPCA can achieve similar or better performance than the state-of-the-art supervised algorithms ICA-FX, MMC, NLDA, CLDA, LDA, KLDA, and SDA. In addition, three models MPCA1, MPCA2, and MPCA3 are comparable with many other PCA-based methods in terms of the classification accuracy.

C. Feature Selection Experiment

In this experiment, we have conducted the experiment on the AR and CMU PIE data sets to demonstrate the effectiveness of our feature selection approach. For comparison, we have implemented two other feature selection methods. One is the filter method, and the other is the wrapper method. In the filter method, we compute Fisher criterion J of each feature v. That is, $J = v^T S_b v/(v^T S_w v)$, where S_b and S_w are the between-class and within-class matrices, respectively [20]. Then, like LDA using the eigenvectors associated with the first eigvenvalues, we select the features corresponding to the first largest J. This method is denoted by FISH. The wrapper method we use is genetic algorithm (GA) [59]. After obtaining three similarity subspaces, we randomly ran FISH and GA feature selection algorithms 10 times on each training subset used in the first and second experiments. Fig. 5 shows the best results of three methods (FISH, GA, and our WMC).

In GA feature selection, there are two important parameters needing to be set. They are generation (i.e., the number of the iterations) G and the number of the individuals T in each population, in which an individual shows a feature set (for details, please refer to [59]). Here, G is set to 100, and T is set to 50. Note that if we use larger G and T, e.g., $G = 300$ and $T = 300$, that can lead to slightly better classification performance in theory (improve about 1% in our experiment), the computational cost of GA is very high. From Fig. 5, we can observe that FISH is worse than WMC

TABLE I
CLASSIFICATION ACCURACIES (MEAN ± STD-DEV PERCENT) ON THE AR DATA SET

Algorithms	*N = 9*	*N = 10*	*N = 11*	*N = 12*	*N = 13*
NN	72.99±1.14	75.44±0.87	77.39±0.74	79.01±0.78	80.76±1.17
1DPCA_L2	73.0±1.14	75.48±0.85	77.43±0.73	79.02±0.77	80.81±1.14
2DPCA_L2	73.02±1.13	75.45±0.86	77.43±0.73	79.05±0.79	80.81±1.15
KPCA	72.99±1.15	75.47±0.86	77.43±0.73	79.02±0.77	80.79±1.15
1DPCA_L1	72.97±1.16	75.46±0.87	77.43±0.74	79.03±0.79	80.81±1.15
2DPCA_L1	73.24±1.36	75.49±0.92	77.94±0.58	79.04±0.80	80.92±1.02
ICA	73.35±1.32	75.85±0.79	77.52±0.58	78.81±0.73	80.46±1.10
LPP	77.07±2.34	80.51±1.96	82.47±1.49	83.89±1.14	86.42±1.60
MPCA1	73.0±1.15	75.42±0.85	77.46±0.72	79.03±0.80	80.81±1.15
MPCA2	72.99±1.14	75.47±0.86	77.44±0.72	79.02±0.78	80.81±1.13
MPCA3	73.01±1.15	75.47±0.86	77.44±0.72	79.02±0.78	80.80±1.14
ICA-FX	89.24±0.24	90.27±0.65	90.60±0.89	91.49±0.42	91.86±0.95
MMC	85.94±1.14	87.90±0.71	89.28±0.79	90.35±1.13	91.26±0.85
NLDA	***92.49±0.64***	***93.18±0.63***	92.80±0.79	92.98±0.65	92.42±1.09
CLDA	87.05±1.53	89.51±0.95	90.84±0.87	91.92±0.94	92.50±0.88
LDA	91.62±0.82	91.45±0.51	90.88±1.0	90.80±0.53	90.61±1.05
KLDA	92.0±1.06	92.11±1.04	91.57±1.75	93.29±1.83	93.16±3.17
SDA	92.0±0.83	92.14±0.60	92.66±0.77	92.56±1.03	93.35±1.25
MPCA	91.42±0.85	92.20±0.53	***92.84±0.38***	***93.39±0.67***	***94.12±0.85***

TABLE II
CLASSIFICATION ACCURACIES (MEAN ± STD-DEV PERCENT) ON THE PIE DATA SET

Algorithms	*N = 5*	*N = 10*	*N = 15*	*N = 20*
NN	49.36±0.91	69.36±1.45	81.01±1.08	86.95±1.18
1DPCA_L2	49.30±0.95	69.33±1.45	81.03±1.09	86.96±1.18
2DPCA_L2	49.36±0.92	69.33±1.45	81.0±1.07	86.92±1.19
KPCA	49.29±0.94	69.33±1.45	81.02±1.09	86.96±1.18
1DPCA_L1	49.15±1.12	68.92±1.32	80.82±1.04	86.95±1.20
2DPCA_L1	49.37±0.92	69.34±1.43	81.0±1.08	86.96±1.18
ICA	49.23±0.89	69.44±1.38	81.19±1.11	87.12±1.10
LPP	71.28±1.28	80.82±1.21	83.02±0.93	82.99±1.54
MPCA1	49.35±0.93	69.35±1.46	81.03±1.08	86.96±1.17
MPCA2	49.32±0.92	69.35±1.45	81.04±1.09	86.96±1.17
MPCA3	49.29±0.94	69.33±1.45	81.02±1.09	86.96±1.18
ICA-FX	63.32±1.80	82.99±0.65	90.07±1.52	92.50±1.03
MMC	76.16±1.37	86.88±0.75	91.89±1.09	94.17±0.63
NLDA	82.71±1.26	89.77±0.96	93.11±0.98	94.16±0.65
CLDA	74.69±1.83	86.23±0.69	91.15±1.16	94.10±0.83
LDA	81.47±1.62	88.74±1.03	92.49±1.13	93.60±0.60
KLDA	***89.59±1.01***	***91.29±3.84***	93.34±2.64	92.90±2.52
SDA	84.67±1.74	89.39±1.32	92.24±1.16	94.22±0.46
MPCA	82.15±0.45	89.53±0.94	***93.49±0.80***	***95.38±0.56***

and GA. Our WMC outperforms GA. In addition, WMC is more efficient than GA here. We take PIE data set as an example. We perform WMC and GA feature selection on the first subset of CMU PIE (i.e., $N = 5$ in the second experiment) one time. Running GA feature selection spends 1197.31 s, whereas WMC needs only 15.94 s (we have performed GA and WMC on an I7 3.4-GHz Windows 7 machine with 16 GB of memory).

D. Clustering

To show the clustering ability of the vectors that yield the MPCA model, we conducted the experiment on the GT face data set. The GT data set contains 50 subjects with 15 images per subject and characterizes several variations such as pose, expression, and illumination [47]. All the images are cropped and resized to a resolution of 60 × 50 pixels.

In the experiment, we implemented the state-of-the-art clustering algorithms including the traditional K-means algorithm [60] and the spectral clustering algorithms. The basic idea of the spectral clustering methods is to cluster points using eigenvectors of matrices derived from the data [61]. In this sense, the algorithm PCA + K-means, that is, K-means clustering in the principal component subspace [62], is one type of the spectral algorithms. For comparison, we implemented our four

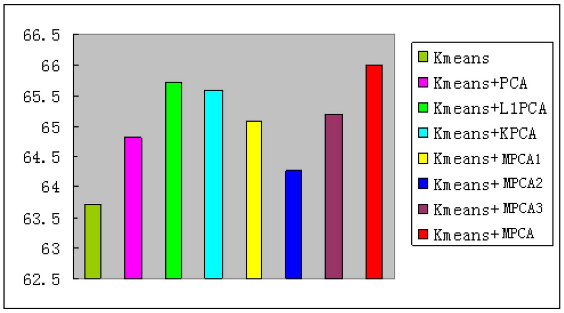

Fig. 6. Clustering performance on GT.

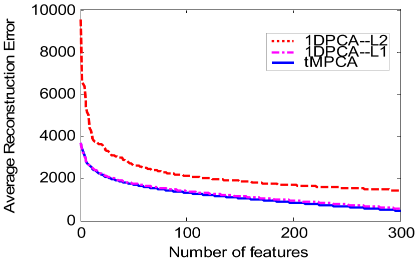

Fig. 7. ARE on the ORL data set.

clustering algorithms: 1) MPCA1 + K-means; 2) MPCA2 + K-means; 3) MPCA3 + K-means; and 4) MPCA + K-means. In the proposed MPCA + K-means algorithm, we first computed three vector sets, respectively, yielding the MPCA1, MPCA2, and MPCA3 subspaces. Then, the vectors in these three sets are orthogonalized to yield a new subspace, denoted by MPCA subspace. Finally, we performed the K-means in the MPCA subspace. All the algorithms used the class centers as the initial cluster centroid positions. We evaluated the clustering performance using the clustering accuracy, which is computed by exploiting the known class labels. We report the highest clustering accuracy of each algorithm, as shown in Fig. 6. We can observe from this figure, our proposed algorithms are better than the typical clustering algorithms.

E. Face Reconstruction

In this experiment, we applied our method to face reconstruction problems to show the representative capability of vectors (i.e., extracted features) in the MPCA implementation. The experiment is conducted on the ORL face data set [47], [57]. Since MPCA is a 1DPCA-based algorithm, for the purpose of fair comparison, we implemented MPCA and the other 1DPCA-based algorithms: 1DPCA_L2 and 1DPCA_L1. We use the method in [7] to compute the average reconstruction error (ARE) of the 1DPCA based algorithms as follows:

$$\text{are}(m)_{1D} = \frac{1}{n}\sum_{i=1}^{n}\left\| x_i - \sum_{j=1}^{m} w_j w_j^T x_i \right\|_2 . \tag{18}$$

Here, n is the number of the total images in the ORL data set ($n = 400$), x_i is the ith 1-D face image expressed as a vector, w_i is the ith extracted feature, and m is the number of features used in the reconstruction.

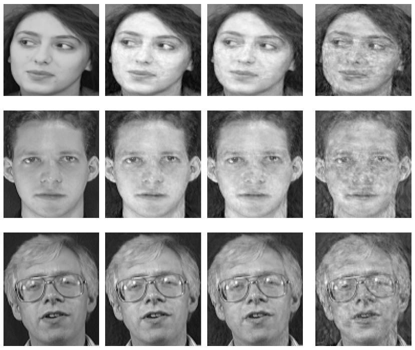

Fig. 8. Original images (in the ORL data set) and reconstructed images. The first column shows the three original images and the second to fourth columns (from left to right) are the images reconstructed by tMPCA, 1DPCA-L1, and 1DPCA_L2, respectively.

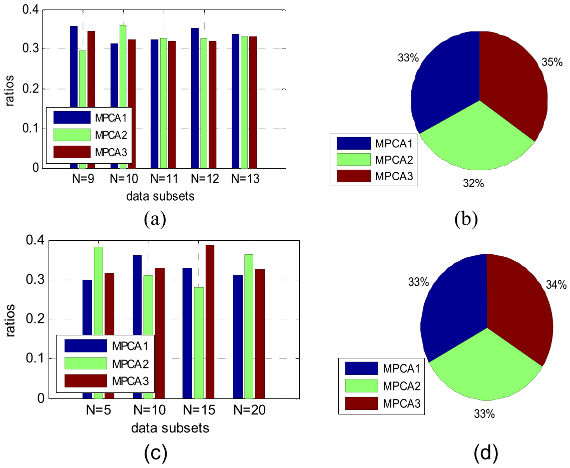

Fig. 9. Components of the optimal features. (a) Ratios on subsets of the AR data set. (b) Average ratio on the AR data set. (c) Ratios on subsets of the PIE data set. (d) Average ratio on the PIE data set.

For the 1DPCA_L2 and 1DPCA_L1 algorithms, we used first 300 extracted features to reconstruct the images. In addition, after orthogonalizing the features of all the three similarity subspaces, we exploit the first 300 orthogonalized features to reconstruct the images in our method. Here, we call this method as tMPCA reconstruction method. Fig. 7 shows the ARE of the above reconstruction algorithms on the ORL data set. From this figure, we can observe that tMPCA is the best algorithm among three 1DPCA-based algorithms. Fig. 8 shows three original images randomly drawn from the ORL data set and their reconstructed images generated by the above three PCA-based reconstruction algorithms. In this figure, the first column shows the three original images and the second to fourth columns (from left to right) are the images reconstructed by tMPCA, 1DPCA-L1, and 1DPCA_L2, respectively. The main reason for powerful reconstruction ability of the proposed approach may be that it can effectively find the most representative feature vectors from three similarity subspaces.

F. Similarity Subspaces in Classification

As demonstrated in Section IV-E, the orthogonalized features drawn from the three similarity subspaces have the powerful capability to represent the data. Moreover, from these features, MPCA can select a number of powerful features that can effectively capture the difference between the samples from different classes. We employ these selected features to classify the samples and achieve very desirable classification results shown in the first two experiments. Actually, if we use more similarity measurements to produce the similarity subspaces, we can obtain more representative features. Therefore, we can select more powerful features and achieve better classification results. In other words, more similarity subspaces may lead to better classification results. Each similarity subspace can make a contribution to the classification task. Fig. 9 shows an example of this case in our experiments.

For the AR and PIE data sets, we randomly ran MPCA 10 times on the subsets that are the same as the ones in the first two experiments. We can obtain the optimal features achieving the best classification results on these subsets. Among these features, we count the number of the features from each similarity subspace. In Fig. 9(a) and (c), we report the ratio of the number of features from each subspace to the total number of the optimal features on each subset. Fig. 9(b) and (d) show the average ratios on the AR and PIE data sets, respectively. From Fig. 9, we observe that the optimal features contain the features from all the three similarity subspaces. This shows that all of these subspaces make a contribution to the classification tasks, although they play different roles in classification. For example, for the AR data set, MPCA3 obtains the highest average ratio of 35%, which shows that the subspace generated by MPCA3 plays more important role than two other subspaces.

V. Conclusion

In this paper, we investigated the relationship between the representative and the discriminative vectors of the data, and modified the PCA algorithm to a novel similarity subspace learning framework MPCA by borrowing the idea of the graph embedding learning. MPCA integrates three subspaces based on the similarity measurements of mutual information, angle information (cosine distance), and kernel distance through a novel feature selection scheme. This scheme based on the weak learning theory can effectively capture sufficient discriminative information. MPCA can achieve desirable classification and clustering results, as well as have a relatively powerful capability to represent the data, as demonstrated in Section IV.

Our MPCA framework is very well suited to learn the high-dimensional data in the case where the scale of training data set is small. Particularly, if each feature vector or transformation axis in the similarity subspaces yields low classification accuracy in a classification algorithm, e.g., the NN classifier, our proposed algorithm MPCA can significantly improve the classification performance of the PCA-based algorithm. The three MPCA models can be applied to both supervised and unsupervised learning scenarios. Among these models, MPCA1 can also be referred to as mutual information PCA that is a novel type of PCA-based method.

The proposed method provides an in-depth understanding of the PCA-based methods, and a new way for modifying the traditional PCA method from the viewpoint of graph embedding learning. The idea of the proposed MPCA can be applied to the other feature extraction approaches. Our similarity subspaces can be replaced by other types of similarity subspaces. Moreover, we can combine the proposed three MPCA models and other linear subspace models, such as LDA, in practice. Our future work is to apply the idea of MPCA to the other linear subspace approaches.

Acknowledgment

The authors would like to thank the editor and the anonymous reviewers for their critical and constructive comments and suggestions.

References

[1] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1222–1228, Sep. 2004.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognit. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.

[3] R. Fisher, "The use of multiple measures in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[4] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[5] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 131–137, Jan. 2004.

[6] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[7] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1672–1680, Sep. 2008.

[8] D. Meng, Q. Zhao, and Z. Xu, "Improve robustness of sparse PCA by L1-norm maximization," *Pattern Recognit.*, vol. 45, pp. 487–497, Jan. 2012.

[9] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, pp. 1–25, 2011.

[10] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst., Man, Cybern., Part B, Cybern.*, vol. 40, no. 4, pp. 1170–1175, Aug. 2010.

[11] B. Schölkopf, A. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[12] S. Y. Huang, Y. R. Yeh, and S. Eguchi, "Robust kernel principal component analysis," *Neural Comput.*, vol. 21, no. 11, pp. 3179–3213, 2009.

[13] Z. Koldovsky, P. Tichavsky, and E. Oja, "Efficient variant of algorithm FastICA for independent component analysis attaining the Cramér-Rao lower bound," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1265–1277, Sep. 2006.

[14] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, no. 2, pp. 443–482, 1999.

[15] R. He, B. G. Hu, W. S. Zheng, and X. Kong, "Robust principal component analysis based on maximum correntropy criterion," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1485–1494, Jun. 2011.

[16] J. Zhao, P. L. H. Yu, and J. T. Kwok, "Bilinear probabilistic principal component analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 492–503, Mar. 2012.

[17] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognit.*, vol. 33, no. 10, pp. 1713–1726, 2000.

[18] M. Zhu and A. M. Martinez, "Subclass discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.

[19] S. Wang, J. Yang, N. Zhang, and C. Zhou, "Tensor discriminant color space for face recognition," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2490–2501, Sep. 2011.

[20] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," *IEEE Trans. Neural Netw.*, vol. 22, no. 7, pp. 1119–1132, Jul. 2011.

[21] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. FisherFaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[22] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller, "Fisher discriminant analysis with kernels," in *Proc. IEEE Signal Process. Soc. Workshop Neural Netw. Signal Process.*, Aug. 1999, pp. 41–48.

[23] H. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Two-dimensional FLD for face recognition," *Pattern Recognit.*, vol. 38, no. 7, pp. 1121–1124, 2005.

[24] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 212–220, Jan. 2007.

[25] S. Chen, H. Zhao, M. Kong, and B. Luo, "2D-LPP: A two-dimensional extension of locality preserving projections," *Neurocomputing*, vol. 70, nos. 4–6, pp. 912–921, 2007.

[26] B. Moghaddam, "Principal manifolds and probabilistic subspaces for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 780–788, Jun. 2002.

[27] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[28] N. García-Pedrajas and D. Ortiz-Boyer, "Boosting random subspace method," *Neural Netw.*, vol. 21, no. 9, pp. 1344–1362, 2008.

[29] S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[30] X. Jiang, "Linear subspace learning-based dimensionality reduction," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 16–26, Mar. 2011.

[31] J. M. Leiva-Murillo and A. Artes-Rodriguez, "Maximization of mutual information for supervised linear feature extraction," *IEEE Trans. Neural Netw.*, vol. 18, no. 5, pp. 1433–1441, Sep. 2007.

[32] K. Torkkola and W. M. Campbell, "Mutual information in learning feature transformations," in *Proc. 17th ICML*, 2000, pp. 1015–1022.

[33] Y. Freund and R. Schapire, "A desicion-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[34] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a mahalanobis metric from equivalence constraints," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 937–965, Sep. 2005.

[35] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *Phil. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

[36] A. N. Gorban and A. Y. Zinovyev, "Principal graphs and manifolds," arXiv:0809.0490 [cs.LG], 2008, DOI: 10.4018/978-1-60566-766-9.

[37] A. N. Gorban and A. Zinovyev, "Principal manifolds and graphs in practice: From molecular biology to dynamical systems," *Int. J. Neural Syst.*, vol. 20, no. 3, pp. 219–232, 2010.

[38] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2002.

[39] Y. Koren and L. Carmel, "Robust linear dimensionality reduction," *IEEE Trans. Visualizat. Comput. Graph.*, vol. 10, no. 4, pp. 459–470, Jul./Aug. 2004.

[40] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY, USA: Wiley, 2006.

[41] R. Cangelosi and A. Goriely, "Component retention in principal component analysis with application to cDNA microarray data," *Biol. Direct*, vol. 2, no. 2, pp. 1–21, 2007.

[42] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[43] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[44] L. Wang, N. Zhou, and F. Chu, "A general wrapper approach to selection of class-dependent features," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1267–1278, Jul. 2008.

[45] M. Sebban and R. Nock, "A hybrid filter/wrapper approach of feature selection using information theory," *Pattern Recognit.*, vol. 35, no. 4, pp. 835–846, 2002.

[46] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[47] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2106–2112, Nov. 2010.

[48] Y. Xu, D. Zhang, Z. Jin, M. Li, and J.-Y. Yang, "A fast kernel-based nonlinear discriminant analysis for multi-class problems," *Pattern Recognit.*, vol. 39, no. 6, pp. 1026–1033, 2006.

[49] J. R. Beveridge, B. A. Draper, J. M. Chang, M. Kirby, and C. Peterson, "Principal angles separate subject illumination spaces in YDB and CMU-PIE," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 351–356, Feb. 2009.

[50] J. Lu, K. Plataniotis, A. Venetsanopoulos, and S. Z. Li, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 166–178, Jan. 2006.

[51] J. Y. Choi, Y. M. Ro, and K. Plataniotis, "Boosting color feature selection for color face recognition," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1425–1434, May 2011.

[52] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.

[53] J. Yang, A. F. Frangi, D. Zhang, and Z. Jin, "KPCA plus LDA: A complete kernel Fisher discriminant framework for feature extraction and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 2, pp. 230–244, Feb. 2005.

[54] N. Kwak and C.-H. Choi, "Feature extraction based on ICA for binary classification problems," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 6, pp. 1374–1388, Nov. 2003.

[55] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Jan. 2006.

[56] R. Duin and M. Loog, "Linear dimensionality reduction via a heteroscedastic extension of LDA: The Chernoff criterion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 732–739, Jun. 2004.

[57] W. H. Yang and D. Q. Dai, "Two-dimensional maximum margin feature extraction for face recognition," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 1002–1012, Aug. 2009.

[58] D. Cai, X. He, and J. Han, "SRDA: An efficient algorithm for large-scale discriminant analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 1–12, Jan. 2008.

[59] S. J. Russell and P. Norvig, *Artificial Intelligence A Modern Approach*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.

[60] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.

[61] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Adv. Neural Inf. Process. Syst.*, vol. 2, no. 1, pp. 849–856, 2002.

[62] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.

Zizhu Fan is currently pursuing the Ph.D. degree in computer science and technology with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China.

He has published more than 20 journal papers. His current research interests include pattern recognition and machine learning.

Yong Xu received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2005.

He is currently a Professor with the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. He has published more than 40 scientific papers. His current research interests include pattern recognition and machine learning.

Wangmeng Zuo received the Ph.D. degree in computer science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2007.

He is currently an Associate Professor with the School of Computer Science and Technology, HIT. He is the author of 30 scientific papers. His current research interests include image modeling and blind restoration, discriminative learning, biometrics, and computer vision.

Jian Yang received the Ph.D. degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002.

He is currently a Professor with the School of Computer Science and Technology, NUST. He is the author of more than 50 scientific papers on pattern recognition and computer vision. His current research interests include pattern recognition, computer vision, and machine learning.

Dr. Yang is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition Letters*, and *Neurocomputing*.

Jinhui Tang received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively.

He is currently a Professor with the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China. He has authored over 80 journal and conference papers. His current research interests include large-scale multimedia search and computer vision.

Dr. Tang served as a Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and *ACM Transactions on Intelligent Systems and Technology*.

Zhihui Lai received the Ph.D. degree in pattern recognition and intelligence system from the Nanjing University of Science and Technology, Nanjing, China, in 2011.

He has been a Research Associate with The Hong Kong Polytechnic University, Hong Kong, since 2010. Currently, he is a Post-Doctoral Fellow with the Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. His current research interests include pattern recognition, image processing, and compressive sense.

David Zhang (F'08) received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1985, and the second Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 1994.

He is currently a Chair Professor with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. He is the author of more than ten books and 200 journal papers.

Dr. Zhang is a fellow of the International Association for Pattern Recognition. He is the Founder and Editor-in-Chief of the *International Journal of Image and Graphics*, and an Associate Editor of more than ten international journals, including the IEEE TRANSACTIONS AND PATTERN RECOGNITION.