Multivariate Pattern Classification based on Local Discriminant Component Analysis

Nan BU and Toshio TSUJI
Department of the Artificial Complex Systems Engineering
Hiroshima University
Higashi-Hiroshima, 739-8527 JAPAN.
E-mail: bu@ieee.org, tsuji@bsys.hiroshima-u.ac.jp

***Abstract*— This paper proposes a novel local discriminant component analysis (DCA) algorithm that is useful for pattern classification of high-dimensional data. Different from most traditional methods, in which feature extractors are usually used prior to a classifier, the proposed method incorporates the feature extraction process into the classifier. Then, a probabilistic neural network is developed based on the idea of local DCA, in which the whole network including the feature extractor and the classifier can be modulated according to a single training criterion, so that features suited to the classification purpose can be extracted. In this paper, a hybrid training algorithm is proposed on the basis of the minimum classification error (MCE) learning. In simulation experiments, benchmark data are used to prove feasibility of the proposed method.**

Keywords-Gaussian mixture model; orthogonal transformations; multivariate analysis; discriminant component analysis

I. INTRODUCTION

Pattern classification is frequently confronted with high-dimensional feature data in applications such as face recognition and text classification [1],[2]. Usually, feature extraction is conducted prior to a classification process, in order to find a compact feature set to avoid exhaustive computation and to reduce statistically redundant/irrelevant attributes to improve classification performance [3]. In feature extraction techniques, original features (d-dimension) are projected into an m-dimensional space, where $m < d$, and the m axes of the reduced feature space are determined according to some optimal criterion, e.g., a reconstruction error in the PCA-based feature extractor [4]. However, since optimal criteria are frequently used for feature extraction, and most of them are not directly related to training criteria of their counterparts for pattern classification, it may not always be possible to extract features in a reduced form containing sufficient discriminant information [5]-[7].

So far, Fisher's linear discriminant analysis (LDA) [4] has also been widely used for feature extraction. The main idea of LDA is to determine a set of discriminant vectors in the subspace by maximizing a ratio of between-class scatter (covariance) to within-class scatter (see [4] for details). This ratio, also referred as the Fisher discriminant function, illustrates linear separability of the classes under consideration, so that the LDA feature extraction scheme suits for finding discriminant features that are desired for classification. Furthermore, it has been proven that imposing an orthogonality constraint on the set of Fisher discriminant vectors would lead to better discrimination [8]. In the literature, a variety of LDA based feature extract schemes have been proposed to improve classification performance on high-dimensional data [7]-[11].

Although these LDA based schemes show promising characteristics of feature extractor for classification tasks, they still suffer from some intrinsic limitations. For instance, since the Fisher discriminant criterion indicates linear separability of the classes, the LDA schemes fail for non-linear problems. To deal with non-linear problems, mixture discriminant analysis (MDA) has been proposed by extending LDA with normal mixtures [12]. Besides the MDA, non-linear discriminant analysis (NLDA) was used, which combines a multilayer perceptron (MLP) with the Fisher discriminant function, trying to take advantage of the approximating properties of MLP [13]. On the other hand, it should be noticed that, in the existing discriminant analysis methods, optimization processes of the feature extractor and the classification part are made separately, and the training criteria are usually different, since training of the classification part always aims to reach a low error probability. It is expected that training the feature extractor and classification part together with a single criterion, e.g., minimizing an error probability, would yield better classification performance [6].

In this paper, we propose a novel feature extraction approach using orthogonal transformation, which projects the original input space into a lower-dimensional space, and a Gaussian mixture model (GMM), which calculates the posterior probabilities for classification. The proposed method, referred as local discriminant component analysis (DCA), uses mixture model of DCA to deal with data of complicated distribution. Based on the idea of local DCA, a probabilistic neural network (NN) is developed. This network combines the feature extraction process with the classification part, and is trained with the minimum classification error (MCE) learning [14]. Introducing the MCE learning is another major difference between the proposed method and those conventional feature extractors based on LDA. Rather than discriminability that is due to the LDA, the proposed method is expected to extract

features, which enable the classification part to realize a low error probability.

The rest of this paper is organized as follows. Section II introduces the conception of the local DCA, and further develops a probabilistic NN based on it. Then, in Section III, a training algorithm for the local DCA based on the MCE learning is proposed. Experimental results for benchmark dataset are presented in Section IV. Finally, Section V gives a conclusion of this paper.

II. Local DCA

A. Mixture of Gaussian Distributions in the Projected Space

The Gaussian mixture model [4] is a commonly used semi-parametric representation for modeling complicated probability distributions. It is based on a linear combination of several components of simple Gaussian distributions. In this paper, each Gaussian distribution is not expressed by the original feature space $\mathscr{S}$, but by a reduced-dimensional space determined according to the local DCA. Suppose, in the model, there are C classes, and each class has K_c Gaussian components. Given the orthogonal transformation matrix for component k in class c as $\mathbf{V}_{c,k} \in \Re^{d\times M_{c,k}}$, $(c = 1, 2, \cdots, C;\ k = 1, 2, \cdots, K_c)$, the $M_{c,k}$-dimensional input vector of the projected space $\mathscr{S}_{c,k}$ is as

$$\mathbf{x}_{c,k} = \mathbf{V}_{c,k}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}_{c,k}), \tag{1}$$

where $\mathbf{x}$ is the input vector expressed in space $\mathscr{S}$, and $\boldsymbol{\mu}_{c,k}$ is the mean vector of component $\{c, k\}$. The covariance matrix $\mathbf{R}_{c,k} \in \Re^{M_{c,k}\times M_{c,k}}$ is given by

$$\mathbf{R}_{c,k} = \frac{1}{N_{c,k}} \sum_{\mathbf{x}\in \mathrm{component}\{c,k\}} \mathbf{x}_{c,k}\mathbf{x}_{c,k}^{\mathrm{T}}. \tag{2}$$

Here $N_{c,k}$ indicates the number of input vectors belonging to the component $\{c, k\}$. With the projected vector $\mathbf{x}_{c,k}$ and covariance matrix $\mathbf{R}_{c,k}$ defined above, the probability density function (pdf) of $\{c, k\}$ can be written as follows

$$\hat{P}(\mathbf{x}|c, k) = (2\pi)^{-\frac{M_{c,k}}{2}} |\mathbf{R}_{c,k}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{x}_{c,k}^{\mathrm{T}}\mathbf{R}_{c,k}^{-1}\mathbf{x}_{c,k}\right]. \tag{3}$$

Note that $\hat{P}(\mathbf{x}|c, k)$ in (3) does not mean the true pdf for component $\{c, k\}$. It indicates the pdf in the discriminant subspace of $\{c, k\}$, and the mark $\hat{}$ is used in this paper for difference. Then, according to the Bayes theory, the posterior probability of $\mathbf{x}$, in a discriminant sense, is defined as

$$\hat{P}(c, k|\mathbf{x}) = \frac{\alpha_{c,k}\hat{P}(\mathbf{x}|c, k)}{\sum_{c'=1}^{C}\sum_{k'=1}^{K_{c'}} \alpha_{c',k'}\hat{P}(\mathbf{x}|c', k')}, \tag{4}$$

$$\hat{P}(c|\mathbf{x}) = \sum_{k=1}^{K_c} \hat{P}(c, k|\mathbf{x}), \tag{5}$$

where $\alpha_{c,k}$ is the mixing coefficient for $\{c, k\}$, and it equals to the prior probability of $\{c, k\}$, $P(c, k)$.

B. Discriminant Analysis based on MCE

Introducing the MCE learning [14], the discriminant function can be defined as

$$d_c(\mathbf{x}) = -\hat{P}(c|\mathbf{x}) + \left[\frac{1}{C-1}\sum_{c',c'\neq c} \hat{P}(c'|\mathbf{x})^{\eta}\right]^{\frac{1}{\eta}}, \tag{6}$$

and the objective function for each input $\mathbf{x}$ is given as:

$$E(\mathbf{x}) = \sum_{c=1}^{C} E_c(\mathbf{x})I(\mathbf{x} \in c), \tag{7}$$

where $I(s) = 1$ if the statement s is true, otherwise 0, and $E_c(\mathbf{x})$ is defined in the form as

$$E_c(\mathbf{x}) = \frac{1}{1 + e^{-\xi d_c(\mathbf{x})}}. \tag{8}$$

The parameters η and ξ are positive constants, and should be predefined. Since (6) provides a precise misclassification measure, and minimizing the objective function (7) will result in the minimum error probability. In the proposed method, the orthogonal transformation matrices, $\mathbf{V}_{c,k}$ $(c = 1, 2, \cdots, C;\ k = 1, 2, \cdots, K_c)$, are modulated by minimizing (7), and the remained $M_{c,k}$ features are assumed to provide the most discriminant information for classification in component $\{c, k\}$.

C. A Probabilistic NN

It is worth noting that the covariance matrix $\mathbf{R}_{c,k}$ can be reduced to a diagonal matrix $\boldsymbol{\Lambda}_{c,k}$ by an orthogonal similarity transformation [15]:

$$\mathbf{R}_{c,k} = \mathbf{Q}_{c,k}\boldsymbol{\Lambda}_{c,k}\mathbf{Q}_{c,k}^{\mathrm{T}}, \tag{9}$$

where $\mathbf{Q}_{c,k}$ is constructed by $M_{c,k}$ eigenvectors of $\mathbf{R}_{c,k}$, $\mathbf{q}_{c,k}^1, \mathbf{q}_{c,k}^2, \cdots, \mathbf{q}_{c,k}^{M_{c,k}}$, and $\boldsymbol{\Lambda}_{c,k}$ has the associated eigenvalues, $\lambda_{c,k}^1, \lambda_{c,k}^2, \cdots, \lambda_{c,k}^{M_{c,k}}$, as its diagonal elements. Replacing $\mathbf{x}_{c,k}$ and $\mathbf{R}_{c,k}$ in (3) with (1) and (9), we have

$$\begin{aligned}\hat{P}(\mathbf{x}|c, k) &= (2\pi)^{-\frac{M_{c,k}}{2}} |\mathbf{R}_{c,k}|^{-\frac{1}{2}} \exp\Big[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_{c,k})^{\mathrm{T}} \\ &\quad \mathbf{V}_{c,k}\mathbf{Q}_{c,k}\boldsymbol{\Lambda}_{c,k}^{-1}\mathbf{Q}_{c,k}^{\mathrm{T}}\mathbf{V}_{c,k}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}_{c,k})\Big].\end{aligned} \tag{10}$$

Let us define a new orthogonal transformation matrix

$$\mathbf{W}_{c,k} = \mathbf{V}_{c,k}\mathbf{Q}_{c,k}, \tag{11}$$

and a new projection of input vector $\mathbf{x}$ on the subspace $\mathscr{S}_{c,k}$

$$\begin{aligned}\mathbf{x}'_{c,k} &= \mathbf{W}_{c,k}^{\mathrm{T}}(\mathbf{x} - \boldsymbol{\mu}_{c,k}) \\ &= \mathbf{W}_{c,k}^{\mathrm{T}}\mathbf{x} - \mathbf{W}_{c,k}^{\mathrm{T}}\boldsymbol{\mu}_{c,k}.\end{aligned} \tag{12}$$

Therefore the mth element of projected vector $\mathbf{x}'_{c,k}$, $x'^{m}_{c,k}$, can be expressed as

$$\begin{aligned}x'^{m}_{c,k} &= W_{c,k}^{m\ \mathrm{T}}\mathbf{x} - W_{c,k}^{m\ \mathrm{T}}\boldsymbol{\mu}_{c,k} \\ &= \boldsymbol{\beta}_{c,k}^{m\ \mathrm{T}}\mathbf{X},\end{aligned} \tag{13}$$

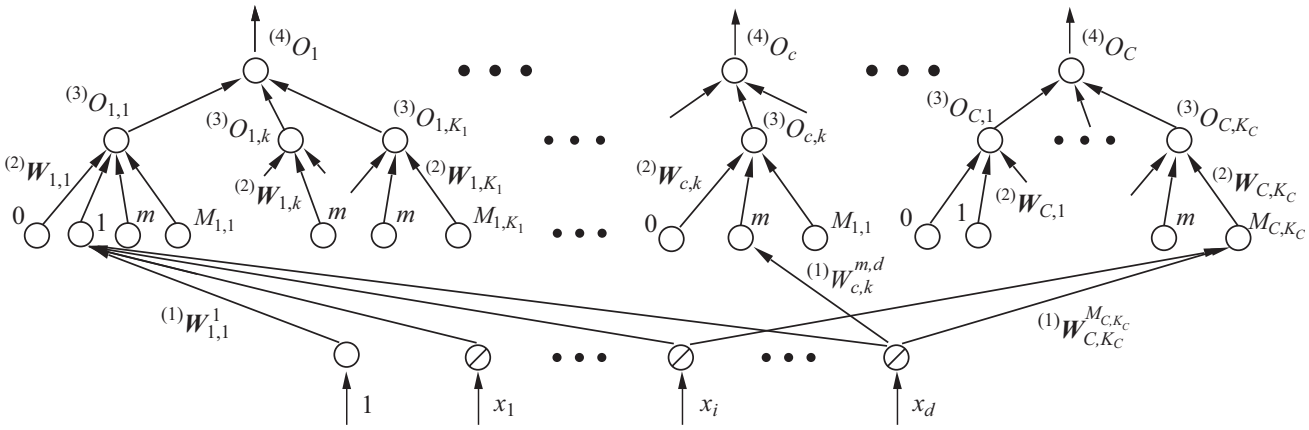

Figure 1. Structure of the probabilistic NN based on local DCA proposed in this paper.

where $W^m_{c,k}$ is the mth row of $\mathbf{W}_{c,k}$; $\mathbf{X}^{\mathrm{T}} = [1, \mathbf{x}^{\mathrm{T}}]$ the modified input vector; and

$$\boldsymbol{\beta}^{m\ \mathrm{T}}_{c,k} = [-W^{m\ \mathrm{T}}_{c,k}\boldsymbol{\mu}_{c,k}, w^{m,1}_{c,k}, \cdots, w^{m,d}_{c,k}]. \quad (14)$$

With (11)-(13), (10) can be simplified as

$$\begin{aligned}\hat{P}(\mathbf{x}|c,k) &= (2\pi)^{-\frac{M_{c,k}}{2}}|\mathbf{R}_{c,k}|^{-\frac{1}{2}}\exp\left(-\sum_{m=1}^{M_{c,k}}\frac{(x'^m_{c,k})^2}{2\lambda^m_{c,k}}\right)\\ &= \exp\left(-\frac{M_{c,k}}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{R}_{c,k}| - \sum_{m=1}^{M_{c,k}}\frac{(x'^m_{c,k})^2}{2\lambda^m_{c,k}}\right)\\ &= \exp\left[\mathbf{X}'^{\ \mathrm{T}}_{c,k}\boldsymbol{\psi}_{c,k}\right]. \end{aligned} \quad (15)$$

where $\mathbf{X}'_{c,k} = (1, (x'^1_{c,k})^2, (x'^2_{c,k})^2, \cdots, (x'^d_{c,k})^2)^{\mathrm{T}}$, and $\boldsymbol{\psi}_{c,k} = (-\frac{M_{c,k}}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{R}_{c,k}|, -\frac{1}{2\lambda^1_{c,k}}, -\frac{1}{2\lambda^2_{c,k}}, \cdots, -\frac{1}{2\lambda^d_{c,k}})^{\mathrm{T}}$. Applying the newly defined vectors, $\boldsymbol{\beta}^m_{c,k}$ and $\boldsymbol{\psi}_{c,k}$, as the weight coefficients, the local DCA algorithm can be developed into a network structure.

It should be mentioned that the orthogonal transformations and modifications of (9)-(15) do not change the characteristics of the projected subspace $\mathscr{S}_{c,k}$ determined by $\mathbf{V}_{c,k}$, but facilitate the development of the local DCA algorithm into a probabilistic NN.

The proposed probabilistic NN is the four-layer feed-forward NN, the structure of which is shown in Fig. 1. The first layer consists of $d+1$ units corresponding to the dimension of $\mathbf{X}$, and the identity function is used for activation of each unit. Let $^{(1)}O_i$ $(i = 0, \cdots, d)$ denote the output of the ith unit in the first layer, we have

$$^{(1)}O_i = \begin{cases} 1, & i = 0 \\ x_i, & i = 1, 2, \cdots, d \end{cases} \quad (16)$$

where $x_i, i = 1, 2, \cdots, d$, is the element of $\mathbf{x}$.

In the second layer, the unit $\{c, k, 0\}$, $(c = 1, \cdots, C; k = 1, \cdots, K_c)$, is the bias unit, and its output $^{(2)}O^0_{c,k} = 1$. On the other hand, the unit $\{c, k, m\}$, $(m = 1, \cdots, M_{c,k})$, receives the output of the first layer weighted by $^{(1)}\mathbf{W}^m_{c,k}$. The input ($^{(2)}I^m_{c,k}$) and the output ($^{(2)}O^m_{c,k}$), for $m \neq 0$, are defined as follows:

$$^{(2)}I^m_{c,k} = \sum_{i=1}^{d} {}^{(1)}O_i\, {}^{(1)}W^{m,i}_{c,k}, \quad (17)$$

$$^{(2)}O^m_{c,k} = (^{(2)}I^m_{c,k})^2. \quad (18)$$

The unit $\{c, k\}$ in the third layer sums up the outputs of the second layer and outputs the posterior probability of each Gaussian component defined in (4). The relationships between the input of unit $\{c, k\}$ in the third layer ($^{(3)}I_{c,k}$) and the output ($^{(3)}O_{c,k}$) are defined as

$$^{(3)}I_{c,k} = \sum_{m=0}^{M_{c,k}} {}^{(2)}O^m_{c,k}\, {}^{(2)}W^m_{c,k}, \quad (19)$$

$$^{(3)}O_{c,k} = \frac{\exp[^{(3)}I_{c,k}]}{\sum_{c'=1}^{C}\sum_{k'=1}^{K_{c'}}\exp[^{(3)}I_{c',k'}]}. \quad (20)$$

The fourth layer consists of C units corresponding to the number of classes. The unit c sums up the outputs of K_c components $\{c, k\}$ in the third layer. The function between the input and the output is described as

$$^{(4)}O_c = {}^{(4)}I_c = \sum_{k=1}^{K_c} {}^{(3)}O_{c,k}, \quad (21)$$

where the output $^{(4)}O_c$ corresponds to the posterior probability of class c (see (5)).

If data are expressed by the GMM with the original dimension d, the parameters in the model include the d-dimensional mean vectors, the d-by-d covariant matrices, and the mixing coefficients in all components. Table I gives the comparison of the numbers of parameters in the GMM of dimension d, the local DCA, and the NN structure. It is clear that, when the reduced dimension $M_{c,k}$ is small, the proposed local DCA algorithm and the corresponding NN would dramatically reduce the parameter numbers, and consequently lighten the computation burden.

TABLE I

COMPARISON OF THE NUMBERS OF PARAMETERS IN THE GMM OF DIMENSION d, THE LOCAL DCA, AND THE NN STRUCTURE.

	Numbers of parameters
GMM of dimension d	$\sum_c \sum_k \frac{1}{2}(d+1)(d+2)$
Local DCA	$\sum_c \sum_k 1 + \frac{1}{2}(d+M_{c,k})(1+M_{c,k})$
NN structure	$\sum_c \sum_k 1 + (d+2)M_{c,k}$

III. TRAINING ALGORITHM FOR LOCAL DCA

In the training procedure, a set of vectors $(\mathbf{x}_1, \cdots, \mathbf{x}_N)$ and the corresponding teacher vector $\mathbf{T}_n = (T_{n1}, \cdots, T_{nc}, \cdots, T_{nC})$ $(n = 1, \cdots, N)$ are used. The teacher vector provides perfect classification, that is, $T_{n\hat{c}} = 1$ for the particular class $\hat{c}$ and $T_{nc} = 0$ for the other classes. The objective function based on MCE is

$$E = \sum_{n=1}^{N} E(\mathbf{x}_n) = \sum_{n=1}^{N} \sum_{c=1}^{C} E_c(\mathbf{x}_n) T_{nc}. \tag{22}$$

A. A Gradient Descent Training Algorithm

For the training of the NN introduced in II-C, the gradient descent of ${}^{(1)}\mathbf{W}_{c,k}^m$ and ${}^{(2)}\mathbf{W}_{c,k}$ can be derived as follows:

$$\begin{aligned}\frac{\partial E(\mathbf{x}_n)}{\partial {}^{(1)}\mathbf{W}_{c,k}^m} &= \sum_{h=1}^{C} \frac{\partial E(\mathbf{x}_n)}{\partial E_h(\mathbf{x}_n)} \frac{\partial E_h(\mathbf{x}_n)}{\partial d_h(\mathbf{x}_n)} \left[\sum_{i=1}^{C} \frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} \right. \\ &\quad \left. \times \sum_{j=1}^{K_i} \frac{\partial \hat{P}(i,j|\mathbf{x}_n)}{\partial \hat{P}(\mathbf{x}_n|c,k)} \right] \frac{\partial \hat{P}(\mathbf{x}_n|c,k)}{\partial \mathbf{X}'_{c,k}(n)} \frac{\partial \mathbf{X}'_{c,k}(n)}{\partial {}^{(1)}\mathbf{W}_{c,k}^m} \\ &= \sum_{h=1}^{C} T_{nh} \xi E_h(\mathbf{x}_n)(1 - E_h(\mathbf{x}_n)) \left[\sum_{i=1}^{C} \frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} \right. \\ &\quad \left. \times (\delta_{i,c} - \hat{P}(i|\mathbf{x}_n)) \right] \hat{P}(c,k|\mathbf{x}_n) \frac{-x'^m_{c,k}(n)}{\lambda_{c,k}^m} \mathbf{X}_n, \end{aligned} \tag{23}$$

and

$$\begin{aligned}\frac{\partial E(\mathbf{x}_n)}{\partial {}^{(2)}\mathbf{W}_{c,k}} &= \sum_{h=1}^{C} \frac{\partial E(\mathbf{x}_n)}{\partial E_h(\mathbf{x}_n)} \frac{\partial E_h(\mathbf{x}_n)}{\partial d_h(\mathbf{x}_n)} \left[\sum_{i=1}^{C} \frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} \right. \\ &\quad \left. \times \sum_{j=1}^{K_i} \frac{\partial \hat{P}(i,j|\mathbf{x}_n)}{\partial \hat{P}(\mathbf{x}_n|c,k)} \right] \frac{\partial \hat{P}(\mathbf{x}_n|c,k)}{\partial {}^{(2)}\mathbf{W}_{c,k}} \\ &= \sum_{h=1}^{C} T_{nh} \xi E_h(\mathbf{x}_n)(1 - E_h(\mathbf{x}_n)) \left[\sum_{i=1}^{C} \frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} \right. \\ &\quad \left. \times (\delta_{i,c} - \hat{P}(i|\mathbf{x}_n)) \right] \hat{P}(c,k|\mathbf{x}_n) \mathbf{X}'_{c,k}(n), \end{aligned} \tag{24}$$

where $\mathbf{X}_n$ and $\mathbf{X}'_{c,k}(n)$ denote the modified input vector and the modified projected vector (see (13) and (15)) corresponding to $\mathbf{x}_n$; $\delta_{i,c}$ is the *Kronecker delta*: $\delta_{i,c} = 1$ when $i = c$, and $\delta_{i,c} = 0$ otherwise. As a result, a backpropagation training algorithm for ${}^{(1)}\mathbf{W}_{c,k}^m$ and ${}^{(2)}\mathbf{W}_{c,k}$ can be derived. Unfortunately, due to the stochastic constraints of the parameters and interdependency between the eigenvalues and the eigenvectors, the weight coefficients cannot be modulated arbitrarily, and convergence of the weights is not confirmed. Alternatively, we propose a hybrid training algorithm for the local DCA in this paper, and the weights in the NN structure are computed according to (11), (14) and (15).

B. A Hybrid Training Algorithm

According to the notation of the local DCA algorithm, modification of the mth transformation vector in matrix $\mathbf{V}_{c,k}$, $V_{c,k}^m$, is given as

$$\Delta V_{c,k}^m = -\gamma \sum_{n=1}^{N} \frac{\partial E(\mathbf{x}_n)}{\partial V_{c,k}^m}, \tag{25}$$

where $\gamma > 0$ is the learning rate, and the vector $\frac{\partial E(\mathbf{x}_n)}{\partial V_{c,k}^m}$ can be derived as follows:

$$\begin{aligned}\frac{\partial E(\mathbf{x}_n)}{\partial V_{c,k}^m} &= \sum_{h=1}^{C} \frac{\partial E(\mathbf{x}_n)}{\partial E_h(\mathbf{x}_n)} \frac{\partial E_h(\mathbf{x}_n)}{\partial d_h(\mathbf{x}_n)} \left[\sum_{i=1}^{C} \frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} \right. \\ &\quad \left. \times \sum_{j=1}^{K_i} \frac{\partial \hat{P}(i,j|\mathbf{x}_n)}{\partial \hat{P}(\mathbf{x}_n|c,k)} \right] \frac{\partial \hat{P}(\mathbf{x}_n|c,k)}{\partial \mathbf{x}_{c,k}(n)} \frac{\partial \mathbf{x}_{c,k}(n)}{\partial V_{c,k}^m} \\ &= \sum_{h=1}^{C} T_{nh} \xi E_h(\mathbf{x}_n)(1 - E_h(\mathbf{x}_n)) \left[\sum_{i=1}^{C} \frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} \right. \\ &\quad \left. \times (\delta_{i,c} - \hat{P}(i|\mathbf{x}_n)) \right] \hat{P}(c,k|\mathbf{x}_n) \\ &\quad \times - \mathbf{x}_{c,k}(n)^{\mathrm{T}} \mathbf{R}_{c,k}^{-1} \frac{\partial \mathbf{x}_{c,k}(n)}{\partial V_{c,k}^m}, \end{aligned} \tag{26}$$

where $\mathbf{x}_{c,k}(n)$ is the projection of the input vector $\mathbf{x}_n$ on the subspace $\mathscr{S}_{c,k}$, and

$$\frac{\partial d_h(\mathbf{x}_n)}{\partial \hat{P}(i|\mathbf{x}_n)} = \begin{cases} -1, & h = i \\ \frac{\hat{P}(i|\mathbf{x}_n)^{\eta-1}}{C-1} \left[\frac{\sum_{l,l\neq i} \hat{P}(l|\mathbf{x}_n)^{\eta}}{C-1} \right]^{\frac{1}{\eta}-1}, & h \neq i \end{cases} \tag{27}$$

$$\frac{\partial \mathbf{x}_{c,k}(n)}{\partial V_{c,k}^m} = [\mathbf{0}, \mathbf{0}, \cdots, \mathbf{x}_n, \cdots, \mathbf{0}]^{\mathrm{T}}. \tag{28}$$

$\frac{\partial \mathbf{x}_{c,k}(n)}{\partial V_{c,k}^m}$ is an $M_{c,k} \times d$ matrix, where $\mathbf{0}$s stand for zero vectors of dimension d, and the vector $\mathbf{x}_n$ is its mth column. Then new $V_{c,k}^m$ at training iteration $t+1$ is modified in the form

$$V_{c,k}^m(t+1) = V_{c,k}^m(t) + \Delta V_{c,k}^m. \tag{29}$$

To keep the basis of each subspace $\mathscr{S}_{c,k}$ be orthonormal, after the transformation vectors are updated, the Gram-Schmidt orthogonalization process [15] is applied. According to the adaptive subspace theory, if the learning rate γ is selected sufficiently small, the orthonormal bases of the subspaces will converge [6],[16].

To complete the training algorithm, we also need modification rules for $\alpha_{c,k}$ and $\boldsymbol{\mu}_{c,k}$. In this paper, these parameters are updated in a way similar to an iterative EM algorithm for mixture models [17]. As $\alpha_{c,k}$ is the prior probability $P(c,k)$, it can be expressed as

$$\alpha_{c,k} = P(c,k) = \sum_{n=1}^{N} P(c,k|\mathbf{x}_n)P(\mathbf{x}_n). \qquad (30)$$

Let us assume that $P(\mathbf{x}_n) = 1/N$ $(n = 1, \cdots, N)$, and substitute the $P(c,k|\mathbf{x}_n)$ with $\hat{P}(c,k|\mathbf{x}_n)$. Then, we get

$$\alpha_{c,k} = \frac{1}{N}\sum_{n=1}^{N} \hat{P}(c,k|\mathbf{x}_n). \qquad (31)$$

A normalization process is used to satisfy the constraint that $\sum \alpha_{c,k} = 1$. Similarly, the update for each mean vector, $\boldsymbol{\mu}_{c,k}$, is given as

$$\boldsymbol{\mu}_{c,k} = \frac{\sum_{n=1}^{N} \hat{P}(c,k|\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \hat{P}(c,k|\mathbf{x}_n)}. \qquad (32)$$

To prevent the mean vectors move out of its corresponding class's range, summations in (32) can be made just over vectors belonging to class c.

Then, details of the proposed hybrid training scheme is as follows:

1) **Step 1** Initialization:
   a) Set the mean vectors of K_c components in class c with K_c input vectors, which are randomly selected from the training data belonging to class c
   b) Initialize the transformation matrices with randomized values, and then perform the Gram-Schmidt process.
   c) Set $\alpha_{c,k}$ with a randomized value, and let $\sum \alpha_{c,k} = 1$.
2) **Step 2** Compute the posterior probability:
   a) Transform input vectors into each subspace of component $\{c,k\}$, and calculate the covariance matrix $\mathbf{R}_{c,k}$, using (1) and (2).
   b) Compute the posterior probabilities for all input vectors, $\mathbf{x}_n (n = 1, 2, \cdots, N)$, according to (3)-(5).
3) **Step 3** Compute the objective function:
   a) Compute the objective function using formulas (6)-(8) and (22).
   b) Stop the training procedure, if iteration number reaches a predefined number or the objective function is smaller than a preset value; otherwise go to Step 4.
4) **Step 4** Update parameters:
   a) Update orthonormal bases using the gradient descent training, then perform the Gram-Schmidt process.
   b) Update $\alpha_{c,k}$ and $\boldsymbol{\mu}_{c,k}$ for each component $\{c,k\}$.
   c) Go to Step 2.

TABLE II
CLASSIFICATION RESULTS ON THE TRAINING SET AND THE TEST SET FOR $M = 2, 3, 4$.

	Training set	Test set
$M = 2$	75.83 ± 6.22	70.50 ± 3.79
$M = 3$	91.13 ± 2.04	87.50 ± 1.84
$M = 4$	90.73 ± 2.92	87.00 ± 3.09

Mean $\pm$ S.D. [%]

IV. EXPERIMENTS

In this section, we conduct experiments to exam the feasibility of the proposed algorithm. The Ionosphere data used in the experiments are taken from the UCI machine learning repository[†].

The Ionosphere data have 34 numerical features, and there are two classes in this dataset. In the experiments, for each class, numbers of instances used for the training set and the test set were 150 and 100, respectively. The numbers of components in each class were set to one. The dimension of each reduced subspace, $M_{c,k}$ $(c = 1, \cdots, C; k = 1, \cdots, K_c)$, was set as $M = 2, 3, 4$. The learning rate γ was 1.0×10^{-4}. The training process is terminated in each experiment, if the iteration number reaches 10^5 or objective function is smaller than 0.1.

Table II depicts the mean values and standard deviations of the classification rates, for ten independent successful trials, on the training set and the test set in the cases where $M = 2, 3, 4$. Although similar results were obtained for $M = 3$ and $M = 4$, it should be remarked that the convergence of training was much difficult when $M = 4$. On the other hand, when the number of dimension was not set appropriately, e.g., $M = 2$, the classification performance degrades remarkably.

The performance of the local DCA was compared with results of four traditional methods reported in [7]. Feature extractions (FEs) used in these methods are the conventional PCA, the parametric eigenvalue-based FE, the nonparametric eigenvalue-based FE, and no feature extraction (Plain). The parametric eigenvalue-based FE is a LDA based method, and the nonparametric eigenvalue-based FE is similar to the idea of MDA. After the feature extraction process, a nearest neighbor classification technique was applied [7]. The comparison results are shown in Table III. When $M = 3$ and 4, the proposed local

[†] http://www.ics.uci.edu/~mlearn/MLRepository.html

TABLE III

COMPARISON OF CLASSIFICATION RATES BETWEEN THE LOCAL DCA AND FOUR COMPARISON METHODS REPORTED IN [7].

		Mean classification rate [%]	The number of features
The proposed local DCA method		70.50	2
		87.50	3
		87.00	4
Comparison methods reported in [7]	PCA	87.20	9
	Parametric FE	84.30	1
	Nonparametric FE	84.40	2
	Plain	84.90	34

DCA method achieved high classification rates. Especially, when $M = 3$, the proposed method outperforms all other methods.

V. CONCLUSION

This paper proposed a novel feature extraction scheme, a local DCA, for multivariate pattern classification. Distinct from the previous studies, the feature extraction process and pattern classification part of the proposed method are wholly merged into one framework, and the parameters are modulated to minimize the MCE criterion, which enables the features extracted to contain discriminant information. Experiments on benchmark dataset proved the feasibility of the proposed method. It is found that with an appropriate dimension number for the reduced space, the proposed method outperforms all other methods used in the comparison.

Since there are still some problems with the convergence of training, in our future research, we would like to improve the convergence properties of the proposed training algorithm for local DCA. It is also interesting to study the properties of the orthonormal bases obtained with local DCA experimentally and theoretically.

REFERENCES

[1] Anil K. Jain, Robert P.W. Duin, and Jianchang Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4-37, 2000.

[2] M-J. Er, S-Q. Wu, J-W. Lu, H-L. Toh, "Face Recognition with Radial Basis Function (RBF) Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 13, No. 3, pp. 697-709, 2002.

[3] A. Verikas, M. Bacauskiene, "Feature Slection with Neural Networks," *Pattern Recognition Letters*, Vol. 23, No. 11, pp. 1323-1335, 2002.

[4] C.M. Bishop, "Neural Networks for Pattern Recognition," Clarendon Press, Oxford, 1995.

[5] B. Lerner, H. Guterman, M. Aladjem, and I. Dinstein, "Feature Extraction by Neural Network Nonlinear Mapping for Pattern Classification," *Proc. of the 13th Int. Conf. on Pattern Recognition*, Vol. 4, pp. 320-324, Vienna, 1996.

[6] Rohit Lotlikar, Ravi Kothari, "Bayes-Optimality Motivated Linear and Multilayered Perceptron-Based Dimensionality Reduction," *IEEE Trans. on Neural Networks*, Vol. 11, No. 2, pp. 452-463, 2000.

[7] A. Tsymbal, S. Puuronen, M. Pechenizkiy, M. Baumgarten, D. Patterson, "Eigenvector-based Feature Extraction for Classification," *Proc. of Int. FLAIRS Conf. on Artificial Intelligence*, pp. 354-358, Pensacola, USA, 2002.

[8] D.H. Foley, J.W. Sammon, "An Optimal Set of Discriminant Vectors," *IEEE Trans. on Computer*, Vol. C-24, pp. 281-289, 1975.

[9] W-Y. Zhao, "Discriminant Component Analysis for Face Recognition," *Proc. of the 15th Int. Conf. on Pattern Recognition*, Vol. 2, pp. 822-825, Barcelona, 2000.

[10] J. Duchene, S. Leclercq, "An Optimal Transformation for Discriminant and Principal Component Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, pp. 978-983, 1988.

[11] Z. Jin, J-Y. Yang, Z-S. Hu, Z. Lou, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, Vol. 34, No. 7, pp. 1405-1416, 2001.

[12] T. Hastie, R. Tibshirani, "Discriminant Analysis by Gaussian Mixtrues," *Journal of the Royal Statistical Society, Series B*, Vol. 58, pp. 155-176, 1996.

[13] C.S. Cruz, J. Dorronsoro, "A Nonlinear Discriminant Algorithm for Feature Extraction and Data Classification," *IEEE Trans. on Neural Networks*, Vol. 9, No. 6, pp. 1370-1376, 1998.

[14] B-H. Juang, S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Signal Processing*, Vol. 40, No. 12, pp. 3043-3054, 1992.

[15] P.M. Cohn, "Algebra," Vol. 1, Second Ed., John Wiley & Sons, 1982.

[16] T. Kohonen, "Self-Organizing Maps," Second Ed., Springer-Verlag, 1997.

[17] M.E. Tipping, C.M. Bishop, "Mixtures of Probabilistic Principal Component Analysers," *Neural Computation*, Vol. 11, No. 2, pp. 443-482, 1999.