Available online at www.sciencedirect.com
SCIENCE DIRECT
Pattern Recognition 39 (2006) 1215–1229
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Class-dependent PCA, MDC and LDA: A combined classifier for pattern classification

Alok Sharma[a,*], Kuldip K. Paliwal[a], Godfrey C. Onwubolu[b]

[a]*Signal Processing Lab, Griffith University, Brisbane, Australia*
[b]*Department of Engineering, University of the South Pacific, Suva, Fiji*

Received 3 June 2005; received in revised form 2 January 2006; accepted 1 February 2006

Abstract

Several pattern classifiers give high classification accuracy but their storage requirements and processing time are severely expensive. On the other hand, some classifiers require very low storage requirement and processing time but their classification accuracy is not satisfactory. In either of the cases the performance of the classifier is poor. In this paper, we have presented a technique based on the combination of minimum distance classifier (MDC), class-dependent principal component analysis (PCA) and linear discriminant analysis (LDA) which gives improved performance as compared with other standard techniques when experimented on several machine learning corpuses.

Keywords: Classification accuracy; Total parameter requirement; Processing time; Class-dependent PCA; LDA

1. Introduction

Humans can easily recognize faces, spoken words, handwritten or printed digits, images and many other things in everyday life. A high school teacher can recognize several of his/her students in a classroom just by looking at their faces, though it would be difficult for him/her to recognize the faces of all the students in his/her school. Thereafter, he/she can also speak out their names or act accordingly once the faces are recognized. This recognition becomes feasible due to some adaptation process in human brain which is a gift of nature to mankind. In other words, if there is a limited number of categories or classes (here, student faces) then recognition performance may be improved; however, the same might not be very efficient if several categories are present.

Over the past several years, study on brain has been conducted and as a result complex mathematical models have been developed with similar functionalities as the brain but at limited extent only. At present, several neural and cognitive systems have evolved which are of immense value in the applications such as banking, multimedia, forensic science, computer vision, remote sensing, image recognition, speech recognition, defect detection in manufacturing, obstacle avoidance in robotics and others. The recognition of objects depend upon several characteristics; for example in face recognition, location of eyes, width and length of nose/mouth, length of eyebrows and complexion etc. are some characters which would give information about a face. The objective of pattern recognition is to identify any given object or pattern and provide some actions or decisions using computers or automated systems.

The pattern recognition problem can be divided into two main categories: (i) supervised classification: where the state of nature for each pattern is known and (ii) unsupervised classification: where the state of nature is unknown and learning is based on the similarity of patterns [1]. In this paper, only supervised pattern classification procedures have been considered. A supervised classification could be subdivided into two main phases namely training phase and testing phase. In the training phase the classifier is made to learn

* Corresponding author. Tel.: +61 679 323 2870; fax: +61 679 323 1538.
E-mail address: sharma_al@usp.ac.fj (A. Sharma).

0031-3203/$30.00
doi:10.1016/j.patcog.2006.02.001

by known categories of patterns and in the classification or the testing phase unknown patterns which were not part of the training dataset are assigned class label of the nearest category of trained patterns.

How well a given pattern classifier/recognizer can classify or make some predefined decisions in the shortest possible time and at the lowest cost would determine its performance. In general, the performance of a classifier depends on several factors. Some of them are [1]:

(i) Number of training samples available to the classifier.
(ii) Generalization ability, i.e. its performance in classifying test patterns which were not used during the training stage.
(iii) Classification error—some measured value based on the incorrect decision of the class labelling of any given pattern.
(iv) Complexity—in some cases the number of features or attributes (dimensions) are relatively larger than the number of training samples usually referred as curse of dimensionality.
(v) Speed—processing speed of training and/or testing phase(s).
(vi) Storage—total amount of parameters required to store after the training phase for classification (testing) purposes.

For a fixed number of training samples in a given classifier model, the performance mainly depends on the generalization ability/capability (classification accuracy), speed and implementation cost (due to storage of information). The number of parameters stored during the training phase that is required in performing classification task (testing), is referred as "total parameters". For a given classifier, we can associate the total parameters to the implementation cost of the classification system and the generalization capability may depend upon the type of parameters (distribution, values etc.) used, which is derived from the training phase of classifiers. The higher the total parameters required for classification task, the costlier the system would be. Another important factor is the speed of the classifier. The higher the computational speed the lower the processing time. We therefore want to reduce the total parameters and processing time and at the same time least sacrifice the classification accuracy. In other words, we search for the optimal classification accuracy or least classification error, involving as minimum total parameters and processing time as possible. This would allow the system to accurately classify/recognize an object as quickly as possible at low cost.

Several classifiers are found today in which minimum distance classifier (MDC) is one of the most economical one. In MDC each class is estimated by a single prototype, usually a centroid. It provides classification at minimal total parameter requirement and computational demand but could be at the price of accuracy. The goal of MDC is to correctly label as many patterns as possible. The MDC method finds centroid of classes and measures distances between these centroids and the test pattern. In this method, the test pattern belongs to that class whose centroid is the closest distance to the test pattern. MDC is used in many pattern classification applications [2–6] including disease diagnostics [7], classification of digit mamographic images [8] and optical media inspection [9].

An alternate way of performing classification is by utilizing linear subspace classifiers [10,11]. Here, each class is represented by its Karhunen–Loéve transform (KLT) [12] or principal component analysis (PCA). The objective of PCA is to find a linear transform for each class using the training patterns for that class in the feature space. This gives class-dependent basis vectors.[1] The first basis vector is in the direction of maximum variance of the given data. The remaining basis vectors are mutually orthogonal and, in order, maximize the remaining variances subject to the orthogonal condition. The principal axes are those orthonormal axes onto which the remaining variances under projection are maximum. These orthonormal axes are given by the dominant eigenvectors (i.e. those with the largest associated eigenvalues) of the covariance matrix. In this classifier, each class is characterized by class-dependent basis vectors and the number of basis vectors used for characterization has to be less than the dimensionality d of the feature space. For more details see Ref. [13].

Linear discriminant analysis (LDA) is a well-known technique for dimensionality reduction. In LDA, the dimensional embeddings are reduced in such a way that the orientation of the projected data of classes on an arbitrary line or space is well-separated from each other. The transformation vectors w are taken so that the criteria J is maximum, where J is the ratio of between-class scatter matrix (S_B) and within-class scatter matrix (S_W) [14]. In a c-class problem the LDA projects from d-dimensional space to $c-1$ or less dimensional space ($\mathbf{R}^d \rightarrow \mathbf{R}^{c-1}$). There are some limitations in applying LDA directly viz. matrix S_W can become singular due to high dimensionality of original feature vectors in comparison with low number of training vectors available. To overcome this limitation, a number of authors have proposed the use of class-independent PCA prior to LDA in the feature extraction stage. Swets and Weng [15] showed two stage PCA plus LDA method where PCA is first used for dimension reduction so as to make S_W non-singular before the application of LDA especially when training samples are scarce. Belhumeur et al. [16] proposed a projection method which is based on LDA and PCA techniques for face recognition. In their technique class-independent PCA is first reduce the original space to $N-c$ (where N is the number of training samples available), and then LDA is applied to reduce the dimension to $c-1$. Zhao et al. [17,18] demonstrated a technique based on the combination of LDA and PCA. A

[1] Note that here we are using the class-dependent PCA for classification. PCA is also used in a class-independent fashion for feature extraction [12].

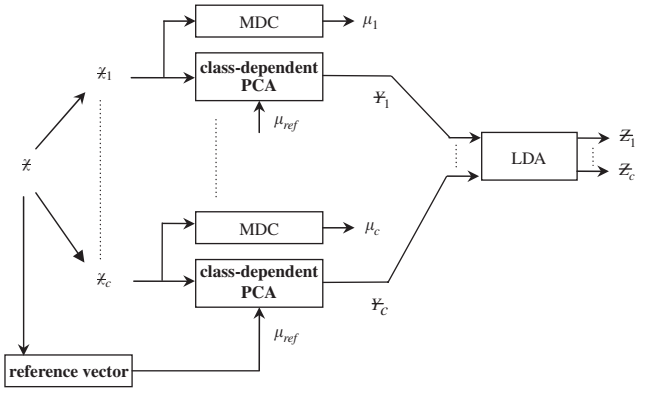

Fig. 1. Framework of MPL classifier.

complete Kernel Fisher discriminant (KFD) was introduced to implement kernel PCA plus LDA strategy by Yang et al. [19] after KFD implementation by Mika et al. [20]. Wu et al. [21] presented a direct LDA method that is applicable to small sample size problems. Jian et al. [22] suggested subspace algorithm for determining the optimal projection for LDA that addressed two LDA problems viz. "small sample size" and "illumination and pose variations". Xiaogang and Xiaoou [23] then presented a unified framework using PCA, LDA and Bayes techniques for face recognition. Ye et al. [24] showed generalized optimization criteria based on pseudoinverse for discriminant analysis to address undersample (S_W being singular) problems. In this paper, we are using class-dependent PCA prior to LDA for classification purposes as shown in Fig. 1.[2]

LDA was applied in face recognition [15,16,23,25,26], in speech recognition [27–29], in speech reading [30] and in optical character recognition [31], and so on.

In this paper we have proposed a unified framework of MDC, class-dependent PCA and LDA techniques. For brevity, we refer to this combination as MPL where "M", "P" and "L" refer to MDC, class-dependent PCA and LDA, respectively.

The strategy of combining classifiers has been previously applied by Xu et al. [32] for handwriting recognition. They have illustrated the combination using some basic classifiers such as Bayesian and kNN, and shown three categories of combination which depend upon the levels of information available from the classifiers. Jacobs et al. [33] suggested supervised learning procedure for systems composed of many separate expert networks. Ho et al. [34] used multiple classifier system to recognize degraded machine-printed characters and words from large lexicons. Tresp and Taniguchi [35] presented modular ways for combining estimators. Woods et al. [36] and Woods [37] presented a method for combining classifiers that uses estimates of each individual classifier's local accuracy in small regions of feature space surrounding a test pattern. Zhou and Imai [38] showed a combination of VQ and multi layer perceptron for Chinese syllables recognition. Alimoglu and Alpaydin [39] used the combination of two multi layer perceptron neural networks for handwritten digit recognition. Kittler et al. [40,41] developed a common theoretical framework for combining classifiers which uses distinct pattern representations. Breukelen van and Duin [42] showed the use of combined classifiers for the initialization of neural network. Alexandre et al. [43] combined classifiers using weighted average after Turner and Gosh [44]. Ueda [45] presented linearly combined multiple neural network classifiers based on statistical pattern recognition theory. Senior [46] used combination of classifiers for fingerprint recognition. Lei et al. [47] demonstrated a combination of multiple classifiers for handwritten Chinese character recognition and Yao et al. [48] used a combination based on fuzzy integral and Bayes method.

The proposed unification (MPL) could reduce the expected distortion $E[\|x-\mu_j\|]$ (due to MDC), mean squared error $E[\|x-\hat{x}\|^2]$ (due to PCA) and maximize the criteria function J on feature space $\mathbf{R}^h$ after the application of class-dependent PCA, where $\hat{x}$ denotes reconstructed vector of x. All the three individual techniques are linearly combined that could help in reducing the classification error. Each constituent technique in MPL may have its own local regions where it performs the best and this could give better performance than individual techniques. To successfully apply LDA technique in the unification, we need to ensure that the scatter matrix S_W does not become singular otherwise it may restrict the direct use of LDA procedure for discriminating features. One way to avoid this situation is to use PCA prior to the application of LDA. We have adopted this two stage PCA and LDA procedure [15–17] which is also known as Fisherface method (an LDA-based technique) [16] and extended the approach by considering class-dependent PCA technique. In Fisherface method, d-dimensional features are firstly reduced to h-dimensional feature space by the application of PCA and then LDA is applied to further reduce features to k dimensions. There are several criteria for determining the value of h [15,17]. One way is to select h such that 95% of the variance present in the original feature is retained [15]. Thus MPL could also be applied under the situation where sample size is scarce.

In this paper we are not proposing any new strategy for combining single classifiers. However, we are using a standard linear combination technique for combining distances from the three classifiers and using the combined distance for classification. The contribution of this paper is as follows:

1. Modified subspace classifier: In Refs. [10,11] as well as in Ref. [13] PCA is taken with respect to origin. We have used subspace classifier with respect to the class centers.
2. We show that out of the three classifiers (MDC, class-dependent subspace classifier (PCA) and class-dependent PCA + LDA) one may give better classifica-

[2] Fig. 1 has been explained in detail in Section 6.

tion result than the others depending on the location of the test vector. Therefore, it makes sense to combine the distances from the three techniques to get better results. In this paper we use linear combination of these three distances.

3. Reference vector (μ_{ref}): In this paper we have shown that class-dependent PCA would produce overlapping of samples in the reduced dimensional space. If LDA is then applied on the obtained transformed samples, it would produce a complex mixture of samples where samples of adjacent class may overlap with each other. This could lead to a poor performance. This defect should be minimized or removed prior to the application of LDA, since if the samples of the adjacent classes are producing overlaps in the transformed space by class-dependent PCA then it would certainly produce overlaps in the transformed space by LDA. We have minimized or removed this problem by introducing a "reference vector" which would prevent the samples of adjacent classes of being overlapped in the transformed space by class-dependent PCA. Then these transformed samples can be further transformed by LDA with fewer errors.

The performance of MPL classifier is compared with MDC, VQ, class-dependent PCA, VQPCA, Fisherface, NN and kNN classifiers and a quantitative analysis of the performance is presented using Sat-Image dataset, TIMIT dataset and Multiple Feature-Digit dataset. The goal of MPL is to label unknown patterns accurately and at the same time maintain total parameter requirement and processing time as low as possible.

The paper is organized as follows: Section 2 focuses on the notations and descriptions used in the paper, Section 3 briefly describes MDC, class-dependent PCA and LDA techniques, Section 4 explains a problem in representing class-dependent PCA before the application of LDA technique, Section 5 deals with the solution of overcoming the problem described in Section 4, Section 6 illustrates a general framework of the MPL classifier, Section 7 deals with the implementation of MPL classifier, Section 8 presents experimentation on machine learning corpuses followed by concluding remarks in Section 9 and acknowledgements.

2. Notations and descriptions

In the remaining discussions $\mathcal{X}$ denotes the d-dimensional set of n training samples in a c-class problem, $\Omega = \{\omega_i : i = 1, 2, \ldots, c\}$ be the finite set of c states of nature or class label where ω_i denotes the ith class label. The set $\mathcal{X}$ is partitioned into c subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_c$ where each subset $\mathcal{X}_i$ belongs to ω_i and consists of n_i number of samples such that

$$n = \sum_{i=1}^{c} n_i.$$

The samples or patterns of set $\mathcal{X}$ can be written as

$$\mathcal{X} = \{x_1, x_2, \ldots, x_n\} \quad \text{where } x_j \in \mathbf{R}^d$$

(d-dimensional hyperplane),

$$\mathcal{X}_i \subset \mathcal{X} \quad \text{and}$$

$$\mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_c = \mathcal{X}.$$

Let $\mathcal{Y}_j$ be h-dimensional transformed samples from $\mathcal{X}_j \in \omega_j$, where $h < d$; then the samples of reduced dimensional set or projected sample set $\mathcal{Y}$ can be depicted as

$$\mathcal{Y} = \{y_1, y_2, \ldots, y_n\} \quad \text{where } y_j \in \mathbf{R}^h$$

(h-dimensional hyperplane),

$$\mathcal{Y}_j \subset \mathcal{Y} \quad \text{and}$$

$$\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup \cdots \cup \mathcal{Y}_c = \mathcal{Y}$$

For convenience, the notations used in the rest of the paper are elaborated as follows:

$L(x)$	class label of test pattern x
μ_j	centroid of subset $\mathcal{X}_j \in \omega_j$
Σ_j	covariance of subset $\mathcal{X}_j \in \omega_j$
Φ_j	$d \times h$ transformation matrix of subset $\mathcal{X}_j \in \omega_j$ (during PCA)
ϕ_i	eigenvector that is a subset of Φ_j
$\hat{\delta}_j$	normalized weighted distance
α or α_i	weighting coefficient
$\hat{x}$	reconstructed pattern of $x \in \mathbf{R}^d$
W	$h \times k$ transformation matrix of set $\mathcal{Y}$ (during LDA)
$\mathcal{Z}_j$	k-dimensional transformed samples from $\mathcal{Y}_j \in \omega_j$ where $k < h$.

3. A review of MDC, class-dependent PCA and LDA classifiers

This section briefly describes three constituent classifiers that are used in designing MPL classifier.

3.1. MDC

The training procedure of MDC technique is simple and straightforward. This is single prototype classifier, i.e. it finds only one feature vector from a given class. MDC estimates a class by its centroid μ_j and store it for later use in the classification task. The centroid can be computed as follows:

$$\mu_j = \frac{1}{n_j} \sum_{x \in \mathcal{X}_j} x \quad \text{for } j = 1, 2, \ldots, c. \tag{1}$$

The storage or total parameter requirement is $d \times c$. In the classification phase, an unknown test pattern is assigned a class label of the stored centroids for which the Euclidean distance is minimum.

3.2. Class-dependent PCA

In class-dependent PCA each class is separately represented by its KLT. For given training samples $x \in \mathcal{X}_j$ in a d-dimensional feature space it will find an orthonormal transformation matrix Φ_j of size $d \times h$ where $h < d$ is a lower-dimensional representation of d-dimensional feature space. The transformation y can be obtained from the vector x and Φ_j as

$$y = \Phi_j^t(x - \mu_j), \tag{2}$$

where $y \in \mathcal{Y}_j \in \mathbf{R}^h$ and μ_j is from Eq. (1). The transformation matrix Φ_j is obtained by minimizing mean squared error $E[\|x - \hat{x}\|^2]$ which turns out to be a generalized eigenvalue problem, i.e.

$$\Sigma_j \phi_i = \lambda_i \phi_i, \tag{3}$$

where $\Phi_j = \{\phi_i : i = 1, 2, \ldots, h\}$, $\phi_i \in \mathbf{R}^d$, $\Sigma_j = E_{x \in \mathcal{X}_j}[(x - \mu_j)(x - \mu_j)^t]$ and λ_i denotes eigenvalues corresponding to ϕ_i.

The eigenvectors $(\phi_1, \ldots, \phi_h)$ of Φ_j should be arranged such that their corresponding eigenvalues are in descending order $\lambda_1 > \lambda_2 > \cdots > \lambda_h$. The direction of first eigenvector ϕ_1 is the direction of maximum variance of given patterns. The second eigenvector (ϕ_2) contains the maximum amount of variance orthogonal to the first one, and so on. The total parameter requirement for class-dependent PCA classifier can be given as

$$\begin{aligned} &\textit{total paramaters} = \textit{centroid\_parameters} \\ &\qquad\qquad + \textit{eigenvector\_parameters}, \\ &\textit{total paramaters} = c \times d + c \times (d \times h) = cd(h + 1). \end{aligned}$$

In the classification phase a test pattern x is assigned the class label for which the reconstruction distance δ_j is minimum. The reconstruction distance can be illustrated as follows:

$$\begin{aligned} &\delta_j = \|x - \hat{x}\| \quad \text{where } \hat{x} = \mu_j + \Phi_j \Phi_j^t (x - \mu_j), \\ &\delta_j = \|x - (\mu_j + \Phi_j \Phi_j^t (x - \mu_j))\| \\ &\quad = \|(I - \Phi_j \Phi_j^t)(x - \mu_j)\|. \end{aligned} \tag{4}$$

3.3. LDA

In LDA the projection is from h-dimensional feature space to k-dimensional feature space where $k < h$ such that the samples or patterns of classes are well-separated. For a c-class problem, assuming $c > 2$ the transformation can be given as

$$z = W^t y \quad \text{where } z \in \mathcal{Z}_j \in \mathbf{R}^k \quad \text{and} \quad y \text{ is from Eq. (2).} \tag{5}$$

The transformation matrix W is computed by maximizing Fisher's criteria $J(W) = |W^t S_B W| / |W^t S_W W|$, where S_B and S_W can be computed from the following expressions:

$$S_B = \sum_{j=1}^{c} n_j (\mu_j - \mu)(\mu_j - \mu)^t \tag{6}$$

and

$$S_W = \sum_{j=1}^{c} \sum_{x \in \mathcal{X}_j} (x - \mu_j)(x - \mu_j)^t.$$

The transformation matrix W is given by [14]

$$S_B w_i = \lambda_i S_W w_i, \tag{7}$$

where $W = \{w_i : i = 1, 2, \ldots, k\}$. The eigenvectors w_i (columns of W) correspond to the eigenvalues λ_i. Since the rank of between-class scatter matrix S_B is $c - 1$ or less, $k \leqslant c - 1$. See Duda and Hart [14] for details.

4. Representation problem with class-dependent PCA

In a c-class problem each class $\mathcal{X}_j \in \mathbf{R}^d$ is separately taken for KLT which yields sample set $\mathcal{Y}_j \in \mathbf{R}^h$. The center of each transformed subset $\mathcal{Y}_j$ (for $j = 1, \ldots, c$) is identical and located at origin in $\mathbf{R}^h$ plane/hyperplane. This may result the samples of $\mathcal{Y}_j$ overlapping with their neighboring classes. If LDA is then applied on the obtained transformed samples, it would produce a complex mixture of samples $\mathcal{Z}_j \in \mathbf{R}^k$ where samples of adjacent classes may overlap with each other. This could lead to poor performance. This defect should be removed prior to the application of LDA, since if the samples of adjacent classes are producing overlaps in the original feature space $\mathbf{R}^h$ then the possibility to get well-separated samples in $\mathbf{R}^k$ feature space would be slim where $k < h$. If only class-dependent PCA is used for representation or compression purposes then this would not be an issue since compressed features can be easily represented back in the original space with some definite errors known as reconstruction error $\|x - \hat{x}\|$.

The center of sample subset $\mathcal{Y}_j$ is a common vector in $\mathbf{R}^h$ feature space; this can be illustrated as follows: from Eq. (2), $y = \Phi_j^t(x - \mu_j)$ where $y \in \mathbf{R}^h$ and $x \in \mathbf{R}^d$ center of $\mathcal{Y}_j$ is simply the sum of Eq. (2), i.e.

$$\begin{aligned} \text{center of } \mathcal{Y}_j &= \frac{1}{n} \sum_{y \in \mathcal{Y}_j} y = \frac{1}{n_j} \sum_{x \in \mathcal{X}_j} \Phi_j^t (x - \mu_j) \\ &= \frac{1}{n_j} \Phi_j^t \left(\sum_{x \in \mathcal{X}_j} x - \sum_{x \in \mathcal{X}_j} \mu_j \right) \\ &= \frac{1}{n_j} \Phi_j^t (n_j \mu_j - n_j \mu_j) \\ &= \mathbf{0} \in \mathbf{R}^h \quad \text{for } j = 1, 2, \ldots, c. \end{aligned}$$

Therefore, it would be helpful to select a reference vector in $\mathbf{R}^d$ feature space that could reduce the probability of

overlapping of adjacent samples in lower-dimensional space $\mathbf{R}^h$. Then LDA can be applied in $\mathbf{R}^h$ space to get well-separated samples in $\mathbf{R}^k$ space. This is described in the next section.

5. Reference vector for discriminant analysis

This section introduces reference vector which is used prior to the applications of class-dependent PCA and LDA techniques. The usage of reference vector with class-dependent PCA and LDA can be viewed in Section 7. The reference vector will be derived from the $\mathbf{R}^d$ sample space which would help in separating features for class-dependent PCA process in $\mathbf{R}^h$ sample space. Therefore, it is required to find a vector in $\mathbf{R}^d$ hyperplane such that its direction and displacement from the origin provides maximum separation of samples of classes $\bar{Y}_j$ in $\mathbf{R}^h$ feature space. In other words, the solution of generalized eigenvalue problem of Eq. (8) will give the required reference vector:

$$S_B v_i = \lambda_i v_i, \tag{8}$$

where eigenvectors v_i are the column vectors of $d \times (c-1)$ rectangular matrix V and λ_i are the corresponding eigenvalues of v_i. Note S_B in Eq. (8) is computed on the sample set $\bar{\chi} \in \mathbf{R}^d$.

The reference vector can be computed from Eq. (8) as

$$\mu_{ref} = \lambda_m v_m, \tag{9}$$

where λ_m is the maximum eigenvalue and v_m is the corresponding eigenvector.

Fig. 2 illustrates the difference between the projections of class-dependent PCA without using the reference vector (Fig. 2a) and class-dependent PCA with reference vector (Fig. 2b) on Sat-Image dataset [49,50,56]. The Sat-Image dataset consists of 36-dimensional samples of six distinct classes. The 36-dimensional patterns are projected onto two-dimensional space and each pattern is represented by a string in the figures which corresponds to their class number.

It can be seen from Fig. 2a that all the samples of classes are drawn around the common center which is the origin of two-dimensional space. If LDA is applied in this reduced two-dimensional space then it may not be able to well separate the samples of each class. On the other hand, the inclusion of reference vector in class-dependent PCA well separates the projected samples of classes in two-dimensional space (Fig. 2b). This would allow the application of LDA with fewer errors.

The maximum eigenvalue obtained by solving Eq. (8) for Sat-Image dataset is very large magnitude ($\lambda_m \approx 10^6$) which is reduced to some reasonable value ($\lambda'_m \approx 10^3$) to get a zoomed-in visualization of projected samples in two-dimensional space.

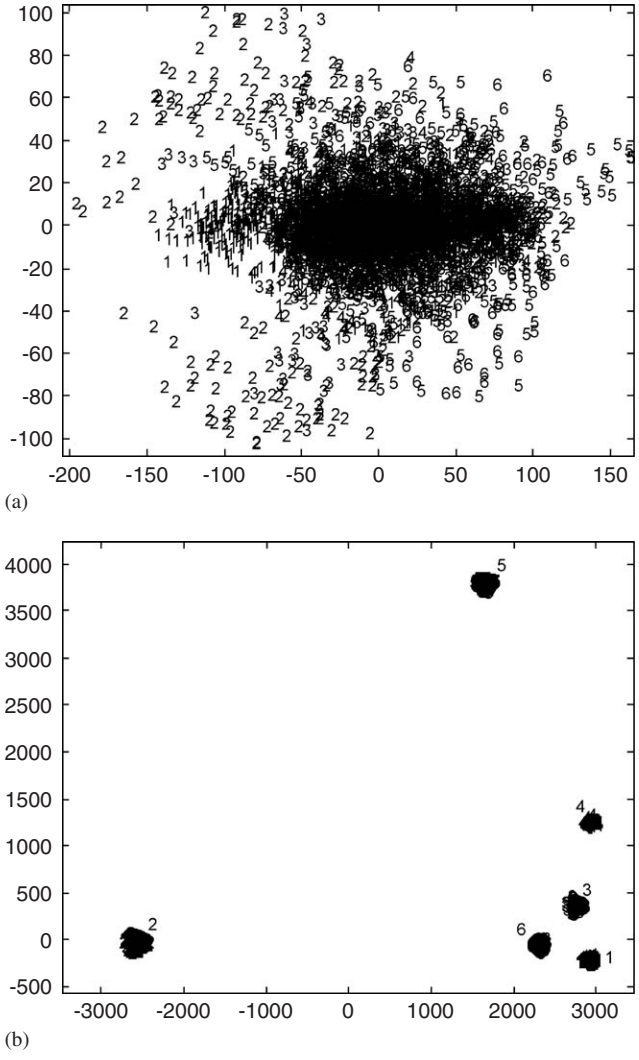

Fig. 2. Projection of 36-dimensional features onto a two-dimensional space using (a) class-dependent PCA, and (b) class-dependent PCA including reference vector.

6. Framework

This section describes the framework of the overall design of MPL classifier. Fig. 1 mainly illustrates the training phase of the classifier. The training sample set $\bar{\chi} \in \mathbf{R}^d$ is first taken for reference vector evaluation procedure. Once the reference vector μ_{ref} is computed it is stored for later use during class-dependent PCA process. The training sample set is a union of the samples of c classes. Therefore, sample set $\bar{\chi}$ can be separated into c subsets $\bar{\chi}_1, \bar{\chi}_2, \ldots, \bar{\chi}_c$ where subset $\bar{\chi}_i$ belongs to the class ω_i. All the subsets are taken for MDC and class-dependent PCA procedures. MDC gives centroid μ_j from the subsets and store it for later use in the classification process. All the same subsets are utilized for class-dependent PCA block including reference vector μ_{ref}. This block would give eigenvector matrix Φ_j which is also stored for later use during the classification phase.

The eigenvector matrix is used to find transformed sample sets $\mathcal{Y}_1, \ldots, \mathcal{Y}_c$ which belong to classes $\omega_1, \ldots, \omega_c$, respectively. Once all the transformed samples are obtained they are then sent to LDA block. This block finds the orientation (direction matrix W which is stored for later use) such that the new projected sample sets $\mathcal{Z}_1, \ldots, \mathcal{Z}_c$ are well separated.

All the classifiers are linearly combined during the classification phase for decision making. The test pattern is assigned the class label of the closest train pattern for which the combined distance is minimal. Section 7.2 depicts the classification phase of MPL classifier in detail.

7. MPL

MPL is a unified framework of MDC, class-dependent PCA and LDA techniques, respectively. It can be subdivided into training phase and testing or classification phase. The combination may help in reducing the expected errors. All the constituent techniques may have their local regions where they may perform the best. MDC will attempt to reduce the expected distortion $E[\|x - \mu_j\|]$, class-dependent PCA will try to minimize the mean squared error $E[\|x-\hat{x}\|^2]$ and LDA will give the orientation for which Fisher's criteria J is maximum i.e. separating the samples of different classes in lower-dimensional space. This approach may improve the generalization capability and at the same time incur low total parameter requirement and low processing time. The application of class-dependent PCA before LDA could also address small sample size problem or singularity problem of matrix S_W. In many cases (e.g. in image recognition or face recognition) the dimension size is very large as compared to the number of feature vectors available. This makes the rank of S_W smaller than the required dimension, making the matrix singular. Several techniques can be found that address small sample size problem e.g. adding a small constant or perturbation to S_W so that it becomes non-singular [17,51], PCA application prior to LDA (also known as Fisherface method) for reducing dimension from d to h such that $h \leqslant$ number of feature vectors [15,16] or the use of pseudoinverse matrix for discriminant analysis [24,52–55].

Our procedure can be categorized under Fisherface method where class-dependent PCA is used instead of PCA and a concept of reference vector is introduced and applied. In addition to class-dependent PCA, MDC is also incorporated in the design. Moreover, we have concentrated more on getting reasonably well generalization capability using minimal possible storage or total parameter requirement and processing time. The performance of a classifier is reasonably well when it gives high classification accuracy at low memory storage and less processing time. A classifier with high accuracy cannot be categorized as an optimum performance classifier if it is extremely slow in giving any results and the memory requirement is extensively high. Thus, we have taken important factors like generalization capability, total parameter requirement and processing time in the consideration while measuring the performance of a classifier. There are several applications of such a classifier for example in obstacle avoidance in robotics where obstacles should be detected at very fast rate and at very low memory storage requirements. In such applications processing time and total parameter requirement are important aspects of classifier performance.

The basic working of the MPL approach can be briefly described by considering a two-class problem. Suppose a three-dimensional feature space of two distinct subsets $\mathcal{X}_1$ and $\mathcal{X}_2$ of class labels ω_1 and ω_2 are given. The MPL approach first finds the center of the distinct subsets separately, i.e. center or centroid is one of the prototypes of the approach. The reason of using centroid as a prototype for MPL is because it may help in reducing the expected distortion and it is computationally inexpensive procedure with minimal storage requirements. The next thing is to find eigenvectors of subsets by using KLT or PCA on each subset separately (class-dependent). The integration of class-dependent PCA would help in reducing high-dimensional space onto a parsimonious data space. The reduction to a lower-dimensional space would be such that the error is minimal in mean squared error sense. Moreover, if the number of samples is scarce then it would not be a disadvantage for MPL since the problem of within-class scatter matrix being singular may be resolved by the use of class-dependent PCA prior to the application of LDA. The application of class-dependent PCA will give eigenvectors in the principal direction and in the secondary directions. These eigenvectors are mutually orthogonal to each other and maximize the variances subject to the orthogonality condition. From the eigenvectors and subsets, MPL will transform three-dimensional feature space to a two-dimensional (say) feature space using reference vector (see Section 5 for details about reference vector). Finally, in MPL algorithm the LDA step is conducted which would maximize the class separation by eliminating redundant components (i.e. minimizing the overlapping of adjacent or neighboring classes in lower-dimensional plane/hyperplane) that may present in a two-dimensional feature space. The application of LDA step in MPL approach will bring a two-dimensional space to a one-dimensional space (say). The LDA step is integrated since it searches for the directions that are efficient for discrimination.

In the classification phase of MPL, an unknown test pattern is allocated a class label (either ω_1 or ω_2) based on the nearest MPL distance measure. This MPL distance measure is derived from the combination of MDC, class-dependent PCA and LDA techniques.

The reason for combining classifiers is simple. A feature or a feature vector could have diverse characteristics. For example, a feature vector may have different structural primitives, physical properties, variation in numerical values [32], etc. All of these diverse characters are then constituted in

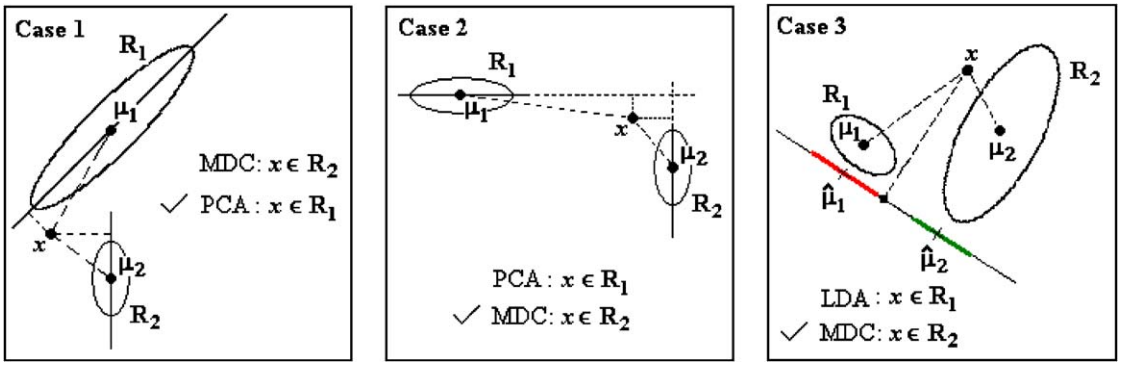

Fig. 3. Three cases where MPL is performing better than the individual techniques.

one vector. This makes it difficult for a single classifier (eg. by MDC or class-dependent PCA or LDA alone) to handle such a diversified feature. If we could combine the different classifiers that could appropriately account for the various forms of a feature then classification performance may improve. To understand this, let us consider the following three cases (Fig. 3) where MPL is performing better than the individual classifiers. In all the three cases the membership of an unknown vector x is to be determined to one of the two regions R_1 and R_2. In case 1, vector x is assigned to R_2 when MDC is used and it is assigned to R_1 when PCA is used. It can be observed that in case 1 MDC is not behaving well than PCA technique since vector x is closer towards R_1 than R_2. On the other hand, in case 2, vector x is assigned to R_1 when using PCA technique which is not performing better than MDC in this particular case. In case 3, a comparison between LDA and MDC is conducted, where vector x is assigned to R_1 when LDA technique is used and it is assigned to R_2 when MDC technique is used. Here MDC is performing better than LDA for the classification of vector x. Similarly several cases can be demonstrated where single classifiers are not performing well. This means that none of the technique could be stated as the best technique. All the techniques have some regions or local regions where they may perform the best and it is also true that on some regions they may not be able to perform satisfactorily well. However, the combination of single classifiers may improve the classification performance. Now considering the same three cases using MPL technique (details about the classification procedure are given in Section 7.2), we can observe that in case 1, x is assigned to R_1 (assuming LDA distance measure is same for both of the regions). Similarly in case 2, x is assigned to R_2 (assuming LDA distance measure is same) and in case 3, x is assigned to R_2 (assuming PCA distance measure is same). This means that MPL is performing better than the other techniques. The various distance used in finding the membership of a vector x are directly measured from the given figure for all the techniques.

The next subsection describes the training phase of MPL classifier.

7.1. Training

The training procedure can be illustrated as follows:

Evaluation of reference vector and MDC:

Step 1: Find between-class scatter matrix and centroid of each class from the training sample set $\mathcal{X}$ (from Eq. (6)):

$$S_B = \sum_{j=1}^{c} n_j(\mu_j - \mu)(\mu_j - \mu)^t,$$

where μ_j can be computed from Eq. (1) and μ can be computed as

$$\mu = \frac{1}{n} \sum_{x \in \mathcal{X}} x.$$

Step 2: Solve generalized eigenvalue problem and find reference vector (from Eqs. (8) and (9)):

$$S_B v_i = \lambda_i v_i,$$

$$\mu_{ref} = \lambda_m v_m.$$

λ_m is the maximum eigenvalue and v_m is the corresponding eigenvector.

Class-dependent PCA step:

Step 1: Find transformation matrix Φ_j from Eq. (3).

Step 2: Project the samples on lower-dimensional space $\mathbf{R}^h$:

$$y = \Phi_j^t(x - \mu_{ref}) \quad \text{where } y \in \mathcal{Y}_j \in \mathbf{R}^h \quad \text{and}$$
$$x \in \mathcal{X}_j \in \mathbf{R}^d \quad \text{for } j = 1, \ldots, c.$$

LDA step:

Step 1: Find transformation W from the generalized eigenvalue problem of Eq. (7):

$$S_B w_i = \lambda_i S_W w_i,$$

where $W = \{w_i : i = 1, 2, \ldots, k\}$, S_B and S_W are computed on sample set $\mathcal{Y}$.

Table 1
List of parameters stored during the training phase which will be required in classification phase with their corresponding size

Parameters	Unit size	Total size for c classes
μ_{ref}	$d \times 1$	d
μ_j	$d \times 1$	dc
Φ_j	$d \times h$	dhc
W	$h \times k$	hk
$\hat{\mu}_j$	$k \times 1$	kc

Step 2: Project the samples on $\mathbf{R}^k$ feature space (Eq. (5)):

$$z = W^t y \quad \text{where } z \in \mathcal{Z}_j \in \mathbf{R}^h \text{ and } y \in \mathcal{Y}_j \in \mathbf{R}^h$$
(it is assumed that $k < h$) for $j = 1, \ldots, c$.

Step 3: Find centroid $\hat{\mu}_j$ of the projected samples $\mathcal{Z}_j$:

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{z \in \mathcal{Z}_j} z. \tag{10}$$

Some parameters are stored during the training phase which will be used in the classification phase. These parameters and their corresponding size are depicted in Table 1.

Thus the total parameter requirement for MPL classifier is the sum of column 3 of Table 1. The next subsection illustrates the classification phase of MPL classifier.

7.2. Classification

The classification procedure of class labelling of an unknown test pattern $x \in \mathbf{R}^d$ is given as follows:

Step 1. Compute the distance δ_j^1 between a test pattern x and the centroid μ_j of class $\mathcal{X}_j \in \omega_j$:

$$\delta_j^1 = \|x - \mu_j\| \quad \text{for } j = 1, 2, \ldots, c.$$

Step 2: Since the addition of reference vector in the classifier design would not affect the orientation of the components of $\mathcal{Y}$ derived by KLT on $\mathcal{X}$, the reconstruction distance can be given by

$$\begin{aligned}\delta_j^2 &= \|x - \hat{x}\| \\ &= \|(I - \Phi_j \Phi_j^t)(x - \mu_j)\| \quad \text{(from Eq. (4))}.\end{aligned}$$

Step 3: Find the projected sample of x due to the reference vector μ_{ref} in h-dimensional space:

$$y_j = \Phi_j^t (x - \mu_{ref}), \quad y_j \in \mathbf{R}^h. \tag{11}$$

The projected pattern onto h-dimensional space would further reduce to k-dimensional space (from Eq. (11)):

$$z_j = W^t y_j \quad \text{where } z_j \in \mathbf{R}^k. \tag{12}$$

Compute the distance between the transformed pattern z_j and transformed centroid $\hat{\mu}_j$ (from Eqs. (10) and (12)):

$$\delta_j^3 = \|z_j - \hat{\mu}_j\|.$$

Step 4: Normalize distances δ_j^1 δ_j^2, δ_j^3 to eliminate the difference in their amplitudes to allow them to contribute equally in decision making.

$$\hat{\delta}_j^1 = \delta_j^1 / \max_{j=1}^{c}(\delta_j^1), \quad \hat{\delta}_j^2 = \delta_j^2 / \max_{j=1}^{c}(\delta_j^2) \quad \text{and}$$
$$\hat{\delta}_j^3 = \delta_j^3 / \max_{j=1}^{c}(\delta_j^3).$$

Step 5: Add distances $\hat{\delta}_j^1$, $\hat{\delta}_j^2$ and $\hat{\delta}_j^3$:

$$\hat{\delta}_j = \alpha_1 \hat{\delta}_j^1 + \alpha_2 \hat{\delta}_j^2 + \alpha_3 \hat{\delta}_j^3,$$

where α_1, α_2 and α_3 are weighting constant in the range [0,1] such that $\sum_{j=1}^{3} \alpha_i = 1$.

Step 6: Find the argument for which the combined distance $\hat{\delta}_j$ is minimum:

$$k = \arg \min_{j=1}^{c} \hat{\delta}_j.$$

Assign the class label ω_k to the test pattern x.

For simplicity we have taken unbiased weighting constant i.e. $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$ (all equal combination) for the classifier design. However, in Fig. 4 we have shown the classification accuracy for 37 different combinations of α's (including all equal combination) using 0.1 weighting interval. The combinations of α are given as follows:

$$\begin{aligned}(\alpha_1, \alpha_2, \alpha_3) = &\underbrace{(0.1, 0.1, 0.8)}_{1}, \underbrace{(0.1, 0.2, 0.7)}_{2}, \\ &\underbrace{(0.1, 0.3, 0.6)}_{3}, \ldots, \underbrace{(0.8, 0.1, 0.1)}_{36} \quad \text{and} \\ &\underbrace{(0.33, 0.3, 0.33)}_{37}.\end{aligned} \tag{13}$$

Three machine learning corpuses have been utilized namely Sat-Image (Fig. 4a) [49,50,56], TIMIT (Fig. 4b) [57] and Multiple Feature-Digit (Fig. 4c) with Zernike-Moments [1,49] in exhibiting classification accuracy vs. dimension-set for all the combination of α (Eq. (13)). The dimension-set refers to $\{h, k\}$ where h denotes reduced dimension obtained by applying class-dependent PCA on initial d-dimensional feature vector and k denotes reduced dimension obtained by applying LDA technique on reduced h-dimensional feature vector. The complete information about all the datasets is given in Section 8.

The bold line in Fig. 4 denotes the all equal weighting combination and slim lines denote the remaining 36 combinations of α as per depicted in Eq. (13). From Fig. 4 it could be observed that there are many weighting combinations for which the classification accuracy is better than the all equal combination and similarly for many combinations the classification accuracy is poor. The classification accuracy obtained by all equal combination is close to optimum in all the three datasets. Empirically, we can take a weighting combination for which optimal performance may

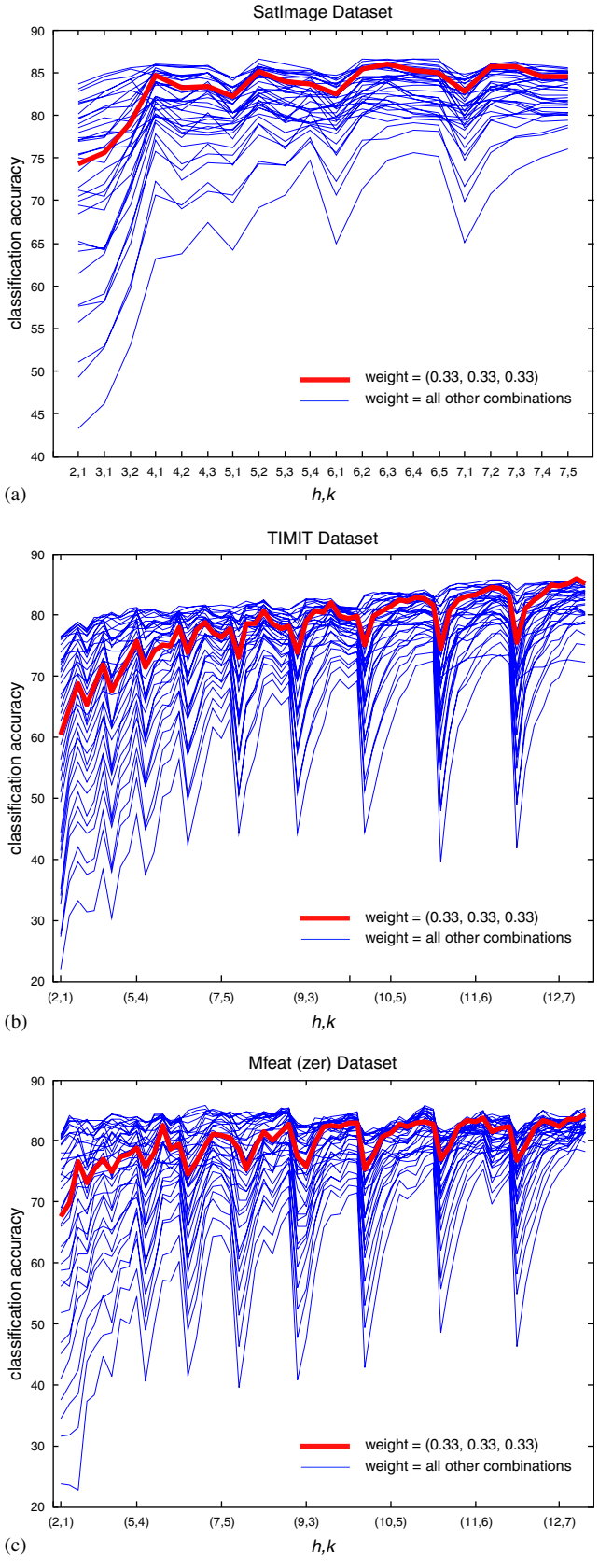

Fig. 4. Classification accuracy vs. dimension-set (h, k) for 37 weighting combinations on (a) Sat-Image dataset, (b) TIMIT dataset, (c) Multiple Feature-Digit (Zernike moments) dataset.

Table 2
Computational complexity of the classification phase of MPL algorithm

No. of step from Section 7.2	Computational complexity
Step 1	$O(dc)$
Step 2	$O(d^2ch)$
Step 3	$O(dch)O(chk) + O(ck) = O(dch)$
Step 4	$O(c)$
Step 5	$O(5)$
Step 6	$O(c)$

be achieved. But since the theoretical framework for the selection of weighting constants has not been developed in this paper, we opted to consider all equal weighting combination for the classifier design. However, Fig. 4 gives an idea that there could be some weighting combinations for which optimal results may be achieved. The interested readers may wish to develop a framework or criteria for the selection of weighting combinations that may depend upon the distribution of a given training dataset and the algorithms that are taken into consideration for the combined classifier design.

7.3. Computational complexity

We would be concentrating on the total computing complexity of a classification session, since the training session can be conducted offline. The computing complexity of each step of Section 7.2 is illustrated in Table 2.

The computing complexity of step 2 is dominating among other steps. Thus, the total computing complexity of the classification phase is estimated to be $O(d^2ch)$.

8. Experimentation

This section demonstrates the performance of the proposed classifier in comparison with MDC, PCA, VQPCA (Euclidean) [58,59], Fisherface, Nearest-neighbor (NN) and k Nearest-neighbor (kNN) [12] techniques. Three sets of machine learning corpuses are utilized namely Sat-Image dataset, TIMIT dataset and Multiple-Feature-Digit dataset to analyze the classifier performance.

The Sat-Image dataset consists of six distinct classes with 36 dimensions or attributes. A sum of 4435 feature vectors is used to train the classifier and a separate set of 2000 vectors is used for verifying the performance of the classifier. From TIMIT corpus a set of 10 distinct vowels are extracted, then each vowel is divided into three segments and each segment is used in getting mel-frequency cepstral coefficients with energy-delta-acceleration (MFCC_E_D_A) feature vectors [60]. A total of 9357 MFCC_E_D_A vectors of dimension 39 for training session and a separate set of 3222 vectors for classification are utilized. The third corpus used is Multiple-Feature-Digit dataset with Zernike moments which consists of 10 distinct classes. A sum of 1500 vectors is used for train-

ing the classifier and a separate set of 500 vectors is used for classification. Each vector has 47 attributes or dimensions.

The performance of the techniques is a measure of classification accuracy as a function of total parameter requirement and processing time. Two plots for each corpus are drawn to exhibit the performance of classifiers. The first plot for each corpus (Figs. 5.1a, 5.2a and 5.3a) shows classification accuracy versus total parameters in logarithmic scale and the second plot (Figs. 5.1b, 5.2b and 5.3b) shows classification accuracy versus processing time.

The following plotting schemes have been adopted for various classifiers:

- MPL, class-dependent PCA and VQPCA: The first value of classification accuracy curve is plotted for which the total parameter requirement is minimum (for e.g. in MPL the starting point of the graph would be when $h=2$ and $k=1$ or in class-dependent PCA when the dimension is reduced to 1). The next value plotted is only that which provided improvement in classification accuracy compared to the previous value. The curve ends at the maximum achievable classification accuracy obtained by a given classifier and thereafter the increase in the total parameters does not improve the classification accuracy any further. This strategy may answer the following question:
  "What is the minimal total parameter requirement and its corresponding processing time to achieve a certain range of classification accuracy?"

  For VQPCA we have opted not to exceed the levels[3] beyond 16 since the classification accuracy does not significantly increase. Moreover, the total parameter requirement and processing time become severely expensive.
- VQ: all four levels 2, 4, 8 and 16 are presented.
- MDC and NN: one unique result is presented.
- kNN: classification accuracy is presented for five values of K, i.e. $K=3$, 5, 7, 9 and 11.
- Fisherface: the LDA dimension is taken as $k=1,\ldots,p-1$ where $p=c$ if $c<h$ or $p=h$ if $h<c$. The value of h is determined by using a criteria presented by Swets and Weng [15].

Figs. 5.1a and b illustrate the performance of classifiers on Sat-Image dataset. It can be observed from Fig. 5.1a that at the beginning, MPL gives classification accuracy of 74.3% at $10^{2.840}$ total parameter requirement and at 3.74 cpu-time in seconds (s). Increasing the total parameter up to $10^{3.200}$ increases the classification accuracy up to 86.1% at 3.77 s. It can be seen that no other presented techniques are producing this classification accuracy at specified total parameters and processing time. It can also be seen that NN and kNN techniques are providing better classification accuracy but their total parameter requirement and processing time are severely expensive i.e. their performance is not very encouraging. It can be observed from Fig. 5.1a that MDC gives minimal total parameter requirement (Fig. 5.1b) but the classification accuracy is quite poor (76.6%). For Fisherface method when using $h=6$ and $k=1,\ldots,5$, the total parameter requirement and processing time are reasonably well though the classification accuracy is not very encouraging.

Next we conducted experiments on TIMIT dataset (Figs. 5.2a and b). It is evident from Figs. 5.2a and b that MPL technique is performing better than all the other techniques including NN and kNN in terms of achieving high classification accuracy at low total parameter requirement and low processing time. The classification accuracy of NN technique is even poorer than MDC, class-dependent PCA, VQ, Fisherface ($h=10$ and $k=1,\ldots,9$) and VQPCA techniques; this means that increasing total parameters does not always help in improving the classification accuracy. The maximum classification accuracy for MPL technique is 86.1% at $10^{3.723}$ using 13.28 s of processing time, whereas the nearest technique to MPL in terms of accuracy is kNN which is giving 78.3% (for $K=11$) at $10^{5.562}$ using 794.08 s of processing time.

Figs. 5.3a and b illustrate several classifiers performance on Multiple-Feature-Digit dataset using Zernike moments. It can be observed from the figures that class-dependent PCA is giving reasonable level of classification accuracy followed by MPL and other techniques. The maximum classification accuracy by class-dependent PCA is 84% at $10^{3.575}$ total parameters and at 0.65 s whereas the maximum classification accuracy by MPL is 84.4% at $10^{3.803}$ total parameters and at 1.38 s. Fisherface ($h=13$ and $k=1,\ldots,9$) is consuming very less processing time but does not provide sufficient classification accuracy. It can also be seen from Fig. 4c that MPL may perform even better than all the presented techniques if a different combination of α is used.

The running time for each step of the MPL technique was also conducted on TIMIT dataset during the training session. It is given here just for an overview of the processing time of each step of the algorithm. The results are illustrated in Table 3. The running time is computed using all the 10 classes and 9357 feature vectors.

In the table, columns 1 and 2 denote dimension reduced by class-dependent PCA (h) and dimension reduced by LDA (k), respectively. Dimension h is shown up to 6 and dimension k is shown up to $h-1$. Columns 3–5 illustrate processing time of each step of MPL during the training session and column 6 shows the total average time in running the algorithm. It can be observed from the table that as h and k increase the running time of the algorithm increases progressively. It is also evident that class-dependent PCA step consumes the maximum running time and increases with h.

It can be concluded from the experiments on several machine learning corpuses that MPL technique produces

[3] Level is the number of disjoint regions in a given class or the number of sub-classes.

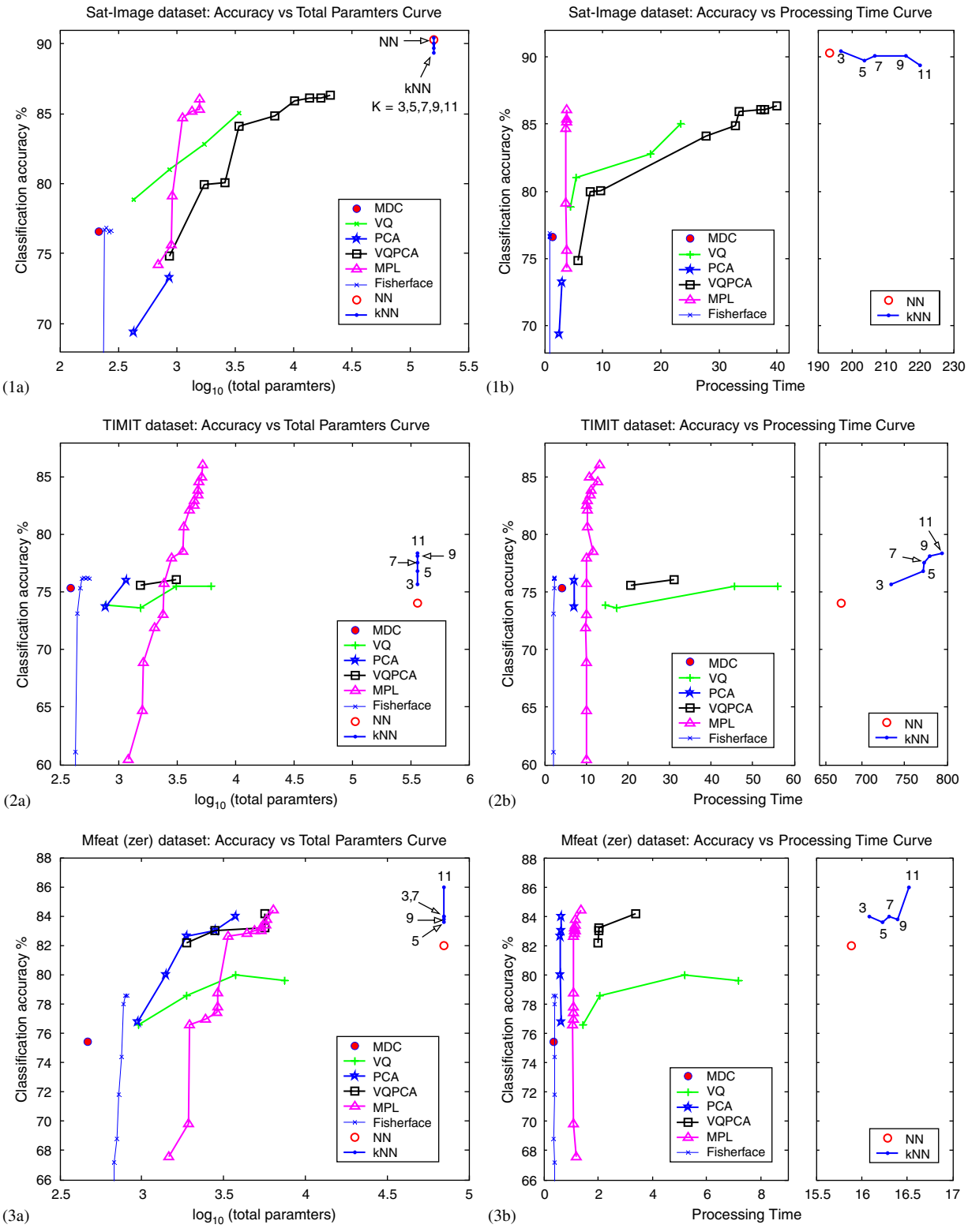

Fig. 5. (1a) Classification accuracy vs. total parameter requirement on Sat-Image dataset; (1b) classification accuracy vs. processing time on Sat-Image dataset; (2a) classification accuracy vs. total parameter requirement on TIMIT dataset; (2b) classification accuracy vs. processing time on TIMIT dataset; (3a) classification accuracy vs. total parameter requirement on Multiple Feature-Digit dataset; and (3b) classification accuracy vs. processing time on Multiple Feature-Digit dataset.

Table 3
Processing time of each step of MPL during the training session on TIMIT dataset

h	k	Evaluation of reference vector and MDC step (s)	Class-dependent PCA step (s)	LDA step (s)	Total average running time (s)
2	1	0.10	3.48	0.03	3.61
3	1	0.08	3.51	0.02	3.62
	2	0.08	3.52	0.02	
4	1	0.07	3.64	0.03	3.70
	2	0.08	3.62	0.02	
	3	0.07	3.54	0.02	
5	1	0.07	3.63	0.03	3.71
	2	0.08	3.59	0.03	
	3	0.09	3.56	0.03	
	4	0.10	3.58	0.03	
6	1	0.07	3.63	0.03	3.72
	2	0.07	3.66	0.03	
	3	0.06	3.56	0.04	
	4	0.05	3.61	0.04	
	5	0.12	3.59	0.03	

promising results in terms of getting classification accuracy in reasonably accepted range and at the same time maintaining minimal total parameter requirement and processing time. This would enable the user to classify a given object accurately and quickly with minimal storage requirements.

We would also like to state here that although MPL is exhibiting better performance overall, it cannot be guaranteed that this technique would produce better performance for all type of data distributions as was seen in the case of Multiple Feature-Digit dataset.

9. Conclusion and future work

The paper presented MPL technique which is a linear combination of MDC, class-dependent PCA and LDA techniques. The performance of the classifiers was measured in terms of classification accuracy as a function of storage and processing time. It was observed that the proposed combination of classifiers provided improved performance over all the other presented techniques. MPL technique on Sat-Image dataset produced maximum classification accuracy of 86.1% at $10^{3.2}$ total parameter requirement and at 3.77 s. NN and kNN techniques also produced higher classification accuracy but the total parameter requirement and processing time were severely expensive. Similarly on TIMIT database, MPL produced the best performance with 86.1% classification accuracy at $10^{3.723}$ total parameter requirement and at 13.28 s of processing time. On the other hand, in Multiple Feature-Digit class-dependent PCA produced best performance among the presented techniques followed by MPL. However, it was noted that MPL could produce better performance if a different combination of α was utilized.

The following questions could be addressed for future work:

- What is the best method of obtaining the subspace dimensions h and k for MPL technique?
- For optimal performance what should be the combination of weighting coefficient α?
- What is the best criterion of combining individual classifiers for MPL?
- Is there any other classifier that can be combined with MPL for further improvement?
- Theoretically on what type of data distributions the algorithm should achieve better performance?

Acknowledgments

The authors would like to acknowledge and show great appreciation to Prof. Erkki Oja of Helsinki University of Technology for kindly enhancing authors' knowledge through his papers. The authors would also like to thank the reviewer for constructive comments which appreciably improved the presentation quality of the paper. This research is partially supported by URC Grant 62005-1361.

References

[1] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 4–37.
[2] V. Di Maio, F. Marciano, Automatic classification of neural spike activity: an application of minimum distance classifiers, Cybern. Syst. 34 (3) (2003) 173–192.

[3] P. Paclik, R.P.W. Duin, Dissimilarity-based classification of spectra: computational issues, Real-Time Imaging 9 (2003) 237–244.
[4] F. Sahin, A radial basis function approach to a color image classification problem in a real time industrial application, Ph.D. Thesis, State University, Virginia, 2000.
[5] P. Datta, D. Kibler, Symbolic nearest mean classifiers, Proceedings of the 14th National Conference on Artificial Intelligence, San Mateo, CA, 1997, pp. 82–87.
[6] S. Griguolo, Pixel-by-pixel clustering for vegetation monitoring, International Conference on Alerte précoce et suivi de l'Environment, Niamey, Niger, 1994.
[7] K. Lewenstein, M. Chojnacki, Minimum distance classifiers in coronary artery disease diagnosing, Modelling in Mechatronics, Kazimierz Dolny, Poland, 2004.
[8] T. Lambrou, A.D. Linney, R.D. Speller, A. Todd-Pokropek, Statistical classification of digital mamograms using features from the spatial and wavelet domains, Medical Image Understanding and Analysis, Portsmouth, UK, 22–23 July, 2002.
[9] D. Toth, A. Condurache, T. Aach, A two-stage-classifier for defect classification in optical media inspection, 16th International Conference on Pattern Recognition (ICPR'02), vol. 4, 2002, pp. 373–376.
[10] E. Oja, Subspace Methods of Pattern Recognition, Research Studies Press, New York, 1983.
[11] E. Oja, J. Parkkinen, On subspace clustering, Seventh International Conference on Pattern Recognition, vol. 2, 1984, pp. 692–695.
[12] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press Inc., Hartcourt, Brace and Jovanovich, New York, 1990.
[13] R.D. Dony, S. Haykin, Image segmentation using a mixture of principal components representation, IEE Proc. Vision, Image Signal Process. 144 (2) (1997) 73–80.
[14] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[15] D.L. Swets, J. Weng, Using discriminative eigenfeatures for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 831–836.
[16] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[17] W. Zhao, R. Chellappa, P.J. Phillips, Subspace linear discriminant analysis for face recognition, Technical Report CAR-TR-914, Center for Automation Research, University of Maryland, College Park, 1999.
[18] W. Zhao, R. Chellappa, N. Nandhakumar, Empirical performance analysis of linear discriminant classifiers, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 164–169.
[19] J. Yang, A.F. Frangi, J.-y. Yang, D. Zhang, J. Zhong, KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2) (2005) 230–244.
[20] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, K.-R. Müller, Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 623–628.
[21] X.-J. Wu, J. Kittler, J.-Y. Yang, K. Messerm S. Wang, A new direct LDA (D-LDA) algorithm for feature extraction in face recognition, Proceedings of the International Conference on Pattern Recognition vol. 4, 2004, pp. 545–548.
[22] H. Jian, P.C. Yuen, C. Wen-Sheng, A novel subspace LDA algorithms for recognition of face images with illumination and pose variations, Proceedings of the International Conference on Machine Learning and Cybernetics, vol. 6, 2004, pp. 3589–3594.
[23] W. Xiaogang, T. Xiaoou, A unified framework for subspace face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (9) (2004) 1222–1228.
[24] J. Ye, R. Janardan, H.P. Cheong, P. Haesun, An optimization criterion for generalized discriminant analysis on undersampled problems, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 982–994.
[25] W. Xiaogang, T. Xiaoou, Using random subspace to combine multiple features for face recognition, Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004, pp. 284–289.
[26] K. Tae-Kyun, K. Hyunwoo, H. Wonjun, K. Seok-Cheol, J. Kittler, Face description based on decomposition and combining of a facial space with LDA, Proceedings of the International Conference on Image Processing, vol. 3, 2003, pp. III-877–III-880.
[27] R. Haeb-Umbach, H. Ney, Linear discriminant analysis for improved large vocabulary continuous speech recognition, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, vol. 1, 1992, pp. 13–16.
[28] O. Siohan, On the robustness of linear discriminant analysis as a preprocessing step for noisy speech recognition, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, vol. 1, 1995, pp. 125–128.
[29] M. Lieb, R. Haeb-Umbach, LDA derived cepstral trajectory filters in adverse environment conditions, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, vol. 2, 2000, pp. II1105–II1108.
[30] G. Potamianos, H.P. Graf, Linear discriminant analysis for speech reading, IEEE Second Workshop on Multimedia Signal Processing, 1998, pp. 221–226.
[31] X. Chen, J. Yang, J. Zhang, A. Waibel, Automatic detection and recognition of signs from natural scenes, IEEE Trans. Image Process. 13 (1) (2004) 87–99.
[32] L. Xu, A. Krzyżak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans Sys. Man Cybern. 22 (3) (1992) 418–435.
[33] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1991) 79–87.
[34] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.
[35] V. Tresp, M. Taniguchi, Combining estimators using non-constant weighting functions, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems, vol. 7, MIT Press, Cambridge, MA, 1995.
[36] K. Woods, K. Bowyer, W.P. Kegelmeyer, Combination of multiple classifiers using local accuracy estimates, IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR '96, 1996, pp. 391–396.
[37] K. Woods, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 405–410.
[38] L. Zhou, S. Imai, Chinese all syllables recognition using combination of multiple classifiers, ICASSP, vol. 6, 1996, pp. 3494–3497.
[39] F. Alimoglu, E. Alpaydin, Combining multiple representations and classifiers for pen-based handwritten digit recognition, International Conference on Document Analysis and Recognition, vol. 2, 1997, pp. 637–640.
[40] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, International Conference on Pattern Recognition, vol. 2, 1996, pp. 897–901.
[41] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[42] M. Breukelen van, R.P.W. Duin, Neural network initialization by combined classifiers, International Conference on Pattern Recognition, vol. 1, 1998, pp. 215–218.
[43] L.A. Alexandre, A.C. Campilho, M. Kamel, Combining independent and unbiased classifiers using weighted average, International Conference on Pattern Recognition, vol. 2, 2000, pp. 495–498.

[44] K. Turner, J. Gosh, Linear and order statistics combiners for pattern classification, in: A. Sharkey (Ed.), Combining Artificial Neural Nets, Springer, Berlin, 1999, pp. 127–161.
[45] N. Ueda, Optimal linear combination of neural networks for improving classification performance, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2) (2000) 207–215.
[46] A. Senior, Combination fingerprint classifier, IEEE Trans. Pattern Anal. Mach. Intell. 23 (10) (2001) 1165–1174.
[47] L. Lei, W. Xiao-Long, L. Bing-Quan, Combining multiple classifiers based on statistical method for handwritten Chinese character recognition, International Conference on Machine Learning and Cybernetics, vol. 1, 2002, pp. 252–255.
[48] M. Yao, X. Pan, T. He, R. Zhang, An improved combination method of multiple classifiers based on fuzzy integrals, World Congress on Intelligent Control and Automation, vol. 3, 2002, pp. 2445–2447.
[49] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998 ⟨http://www.ics.uci.edu/∼mlearn⟩.
[50] D. Michie, D.J. Spiegelhalter, C.C. Taylor (Eds.), Machine Learning, Neural and Statistical Classification, Ellis Horwood, Chichester, 1994.
[51] X.S. Zhou, T.S. Huang, Small sample learning during multimedia retrieval using BiasMap, Proceedings of the IEEE CS Conference on Computer Vision and Pattern Recognition, vol. 1, 2001, pp. 11–17.
[52] R.P.W. Duin, Small sample size generalization, in: G. Borgefors (Ed.), SCIA'95, Proceedings of the Ninth Scandinavian Conference on Image Analysis, vol. 2, 1995, pp. 957–964.
[53] M. Skurichina, R.P.W. Duin, Stabilizing classifiers for very small sample size, Proceedings of the International Conference on Pattern Recognition, vol. 2, 1996, pp. 891–896.
[54] M. Skurichina, R.P.W. Duin, Regularization of linear classifiers by adding redundant features, Pattern Anal. Appl. 2 (1) (1999) 44–52.
[55] S. Raudys, R.P.W. Duin, Expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix, Pattern Recognition Lett. 19 (1998) 385–392.
[56] C. Feng, A. Suntherland, S. King, S. Muggleton, R. Henry, Comparison of machine learning classifiers to statistics and neural networks, AI & Stats Conference, 1993, pp. 41–52.
[57] S.G. Garofalo, L.F. Lori, F.M. William, F.G. Jonathan, P.S. David, D.L. Nancy, The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM, NIST, 1986.
[58] A. Sharma, K.K. Paliwal, G.C. Onwubolu, Pattern classification: an improvement using combination of VQ and PCA based techniques, Am. J. Appl. Sci. 2 (10) (2005) 1445–1555.
[59] N. Kambhatla, Local models and Gaussian mixture models for statistical data processing, Ph.D. Thesis, Oregon Graduate Institute of Science and Technology, 1996.
[60] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland, The HTK Book Version 3.2, Cambridge University, Cambridge, England, 2002.