Journal of VLSI Signal Processing 32, 19–28, 2002

A Modified Minimum Classification Error (MCE) Training Algorithm for Dimensionality Reduction

XUECHUAN WANG AND KULDIP K. PALIWAL
School of Microelectronic Engineering, Griffith University, Brisbane, QLD 4111, Australia

Received April 30, 2001; Revised September 24, 2001; Accepted November 19, 2001

Abstract. Dimensionality reduction is an important problem in pattern recognition. There is a tendency of using more and more features to improve the performance of classifiers. However, not all the newly added features are helpful to classification. Therefore it is necessary to reduce the dimensionality of feature space for effective and efficient pattern recognition. Two popular methods for dimensionality reduction are Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA). While these methods are effective, there exists an inconsistency between feature extraction and the classification objective. In this paper we use Minimum Classification Error (MCE) training algorithm for feature dimensionality reduction and classification on Daterding and GLASS databases. The results of MCE training algorithms are compared with those of LDA and PCA.

Keywords: MCE, LDA, PCA, dimensionality reduction, speech recognition

1. Introduction

One possible way of improving the performance of a pattern recognition system is to use more number of features; i.e., increase the dimensionality of the feature space. The increase in feature dimensionality, however, causes in practice a number of problems. For example, it increases the computational cost and memory requirements. Also, more data is needed for training the pattern recognizer. If only a limited amount of training data is available, the increase in dimensionality makes generalization to test data poorer. Furthermore, the performance of a recognizer is not always enhanced by every newly added feature. Brunzell and Eriksson [1] have shown that the classification results improve when the dimensionality is reduced for some datasets. To solve these problems, it is necessary to reduce the dimensionality of the feature space. A number of dimensionality reduction algorithms have been proposed in the literature to obtain compact feature sets. These methods can be grouped into two categories: feature selection methods and feature extraction methods.

Feature selection methods select features by assigning each feature a score on some basis and choosing the best ranked features to make up a new vector for recognition. The common measures for ranking features are recognition rate, F-ratio and discriminative feature selection measure. Paliwal [2] has investigated the use of the first two measures for feature selection. The third measure is proposed and studied by Bocchieri and Wilpon [3]. However, the recognition results obtained using the feature selection methods depend heavily on the ranking measure used. Since all of these measures have ignored correlations between features, the ranking may not be optimal and the selected features are highly dependent on the choice of the original feature set [4].

Feature extraction methods differ from feature selection methods in that the new features are linear combinations of original features. Thus, the reduced feature set contains all the elements of original feature vectors so that most of the information can be retained. Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) are the two basic feature extraction methods. Both of them reduce the dimensionality of features by projecting the original feature vector into a new subspace through a transformation. But they optimize the transformation with different intentions. PCA optimizes the transformation by finding the largest variations in the original feature space [4–7]. LDA pursues the largest ratio of between-class variation and within-class variation when projecting the original feature to a subspace [8, 9]. A pattern classifier is normally designed to minimize the pattern classification errors as criterion. But, LDA as well as PCA reduce the dimensionality of the feature space using criteria that are different from the minimum classification error criterion. This causes inconsistency between feature extraction and the classification stages of a pattern recognizer [10, 11]. Consequently, the performance of classifiers may be degraded in the reduced feature space. A direct way to overcome this short-coming is to use consistent criteria in both feature extractors and classifiers. In this case, Minimum Classification Error (MCE) training algorithm can be a good choice, since it provides a framework which allows the minimum classification error criteria to be embedded into the feature extraction and pattern classification stages simultaneously [10–13]. Thus, MCE training algorithm achieves minimum classification error directly when extracting features. This direct relationship has made MCE training algorithm widely popular to a number of pattern recognition applications, such as dynamic time-wrapping based speech recognition [14, 15] and HMM based speech and speaker recognition [16–18].

In this paper, we investigate the performance of MCE training algorithm in feature dimensionality reduction and compare its classification performance with that of LDA and PCA. We propose a modified version of the MCE training algorithm and show how it improves the recognition performance in the context of dimensionality reduction.

This paper is organized as follows: Section 2 provides a brief description of the LDA and PCA methods for dimensionality reduction. Section 3 provides a brief review of the MCE training algorithm and describes its use for dimensionally reduction. The modified MCE training algorithm proposed in this paper is then described in this section. In Section 4, the experiments conducted to examine and compare the performance of MCE training algorithms in feature dimensionality reduction are described.

2. Feature Dimensionality Reduction Methods

2.1. General Method for Dimensionality Reduction

The basic way of reducing feature dimensionality is to project the feature vectors into a lower dimensional space through a linear transformation $T_{m\times p}$, where $m < p$, m and p are the dimensionalities of transformed and original feature vectors, respectively. Then the linear transformation T is optimized with respect to some objective criterion. A number of algorithms have been proposed to seek the optimized T with different optimization criteria. LDA and PCA are two most popular ones among them. Briefly speaking, LDA optimizes T by maximizing the ratio of between-class variation and within-class variation; while PCA obtains T by searching for the directions that have the largest variations. LDA is often preferred in practice, as it projects the feature vectors along the directions in which classes have the largest separation while PCA has no intension of discriminating the classes. In the following subsections we will give a detailed discussion of each of them.

2.2. LDA for Dimensionality Reduction

The goal of Fisher's linear discriminant is to well separate the classes by projecting classes' samples from a p-dimension space onto a finely oriented line. For a K-class problem, at most $m = \min(K - 1, p)$ different lines will be involved. Thus the projection is from a p-dimensional space to a m-dimensional space.

Suppose we have K classes, $\chi_1, \chi_2, \ldots, \chi_K$. Let the ith observation vector from the χ_j be x_{ji}, where $j = 1, \ldots, K$, $i = 1, \ldots, N_j$ and N_j is the number of observations from *class* j. Then the sample mean vector μ_j and the covariance matrix S_j of class j are given by:

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_{ji} \tag{1}$$

and

$$S_j = \frac{1}{N_j} \sum_{i=1}^{N_j} (x_{ji} - \mu_j)(x_{ji} - \mu_j)^T \tag{2}$$

The within-class covariance matrix S_W is given by:

$$S_W = \sum_{j=1}^{K} S_j \tag{3}$$

The between-class covariance matrix is defined as:

$$S_B = \sum_{j=1}^{K} N_j(\mu_j - \mu)(\mu_j - \mu)^T \tag{4}$$

where $\mu = \frac{1}{N}\sum_{j=1}^{K} N_j\mu_j$ is the mean of all samples and $N = \sum_{j=1}^{K} N_j$. The projection from a p-dimensional space to a m-dimensional space is accomplished by m discriminant functions:

$$y_s = T_s^T x \quad s = 1, 2, \ldots, m. \tag{5}$$

Equation (5) can be re-written in matrix form:

$$y = T^T x \tag{6}$$

Then the corresponding means and covariance matrices of y are defined as:

$$\tilde{\mu}_j = \frac{1}{N_j}\sum_{i=1}^{N_j} y_{ji} \tag{7}$$

$$\tilde{\mu} = \frac{1}{N}\sum_{j=1}^{K} N_j\tilde{\mu}_j \tag{8}$$

$$\tilde{S}_W = \sum_{j=1}^{K}\sum_{i=1}^{N_j}(y_{ji} - \tilde{\mu}_j)(y_{ji} - \tilde{\mu}_j)^T \tag{9}$$

and

$$\tilde{S}_B = \sum_{j=1}^{K} N_j(\tilde{\mu}_j - \tilde{\mu})(\tilde{\mu}_j - \tilde{\mu})^T \tag{10}$$

It is straightforward to show that:

$$\tilde{S}_W = T^T S_W T \tag{11}$$

and

$$\tilde{S}_B = T^T S_B T \tag{12}$$

Fisher's linear discriminant is then defined as the linear functions $T^T x$ for which the criterion function

$$J(T) = \frac{|\tilde{S}_B|}{|\tilde{S}_B|} = \frac{T^T S_B T}{T^T S_W T} \tag{13}$$

is maximum.

It can be shown that the solution of the above equation for T is in fact the matrix of the leading m eigenvectors of $S_W^{-1} S_B$ [8, 19].

2.3. PCA for Dimensionality Reduction

PCA is a well-established technique for dimensionality reduction [5]. It is based on the assumption that most information about the classes is contained in the directions along which the variations are the largest. The most common derivation of PCA is in terms of a standardised linear projection which maximises the variance in the projected space [7]. For a given p-dimensional data set χ, the m principal axes $T_1, T_2, \ldots, T_m$, where $1 \leq m \leq p$, are orthonomal axes onto which the retained variance is maximum in the projected space. Generally, $T_1, T_2, \ldots, T_m$ can be given by the m leading eigenvectors of the sample covariance matrix $S = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^T(x_i - \mu)$, where $x_i \in \chi$, μ is the sample mean and N is the number of samples, so that:

$$ST_i = \lambda_i T_i \quad i \in 1, \ldots, m \tag{14}$$

where λ_i is the ith largest eigenvalue of S. The m principal components of a given observation vector $x \in \chi$ are given by:

$$y = [y_1, \ldots, y_m] = \left[T_1^T x, \ldots, T_m^T x\right] = T^T x \tag{15}$$

The m principal components of x are then uncorrelated in the projected space. In multi-class problems, the variations of data are determined on a global basis [6], that is, the principal axes are derived from a global covariance matrix:

$$\hat{S} = \frac{1}{N}\sum_{j=1}^{K}\sum_{i=1}^{N_j}(x_{ji} - \hat{\mu})(x_{ji} - \hat{\mu})^T \tag{16}$$

where $\hat{\mu}$ is the global mean of all the samples, K is the number of classes, N_j is the number of samples in class j, $N = \sum_{j=1}^{K} N_j$ and x_{ji} represents the ith observation from class j. The principal axes $T_1, T_2, \ldots, T_m$ are therefore the m leading eigenvectors of $\hat{S}$:

$$\hat{S}T_i = \hat{\lambda}_i T_i \quad i \in 1, \ldots, m \tag{17}$$

where $\hat{\lambda}_i$ is the ith largest eigenvalue of $\hat{S}$.

An assumption made for dimensionality reduction by PCA is that most information of the observation vectors is contained in the subspace spanned by the first m principal axes, where $m < p$. Therefore, each original data vector can be represented by its principal component vector:

$$y = T^T x \tag{18}$$

where $T = [T_1, \ldots, T_m]$ is a $p \times m$ matrix.

The merit of PCA is that the extracted features have the minimum correlation along the principal axes. On the other hand, there are some defects that reside in PCA. First, as mentioned in [4], PCA is a scale-sensitive method, i.e., the principal components may be dominated by the elements with large variances. Another problem with PCA is that the directions of maximum variance are not necessarily the directions of maximum discrimination since there is no attempt to use the class information, such as the between-class scatter and within-class scatter.

3. MCE Training Algorithm

3.1. A Brief Review

MCE training algorithm is a type of discriminant training algorithm. It is proposed to mend the shortcomings of traditional discriminant training. As pointed out by Juang and Katagiri [11], traditional discriminant training algorithms are inadequate in that the decision rule in classification does not appear in the overall criterion functions and there is an inconsistency between the criterion function and the minimum classification error objective. MCE training algorithm bridges this gap by introducing a classification measure, in which the decision rule is embedded, into the overall criterion functions.

Consider an input vector x, the classifier makes its decision by the following decision rule:

$$x \in Class\ k \quad \text{if } g_k(x, \Lambda) = \max_{\text{for all } i \in K} g_i(x, \Lambda) \tag{19}$$

where $g_i(x, \Lambda)$ is discriminant function of x to class i, Λ is the parameter set and K is the number of classes. This criterion can be rewritten as:

$$x \in Class\ k \quad \text{if } g_k(x, \Lambda) - \max_{\text{for all } i \neq k} g_i(x, \Lambda) > 0 \tag{20}$$

Thus, higher is the value of the function $g_k(x, \Lambda) - \max_{\text{for all } i \neq k} g_i(x, \Lambda)$, more reliable is the classification of x into class k. This means that we can use negative of this function as a measure of misclassification of x into class k. But this form is not very suitable since it is not differentiable, and we need to differentiate it for designing a classifier with minimum misclassification error. In [11], a modified version of this function (which is differentiable) is introduced as a misclassification measure. For the kth class, it is given by

$$d_k(x, \Lambda) = -g_k(x, \Lambda) + \left[\frac{1}{N-1} \sum_{\text{for all } i \neq k} (g_i(x, \Lambda))^\eta \right]^{1/\eta}, \tag{21}$$

where η is a positive number and $g_k(x, \Lambda)$ is the discriminant of observation x to its known class k. When η approaches ∞, it reduces to

$$d_k(x, \Lambda) = -g_k(x, \Lambda) + g_j(x, \Lambda), \tag{22}$$

where class j has the largest discriminant value among all the classes other than class k. Obviously, $d_k(x, \Lambda) > 0$ implies misclassification, $d_k(x, \Lambda) < 0$ means correct classification and $d_k(x, \Lambda) = 0$ suggests that x sits on the boundary. The loss function is then defined as a monotonic function of misclassification measure. Sigmoid function is often chosen since it is a smoothed zero-one function suitable for gradient descent algorithm. The loss function is thus given as:

$$l_k(x, \Lambda) = f(d_k(x, \Lambda)) = \frac{1}{1 + e^{-\xi d_k(x, \Lambda)}} \tag{23}$$

where $\xi > 0$. For a training set χ, the empirical loss is defined as:

$$L(\Lambda) = E\{l_k(x, \Lambda)\} = \sum_{k=1}^{K} \sum_{i=1}^{N_k} l_k\left(x^{(i)}, \Lambda\right) \tag{24}$$

where N_k is the number of samples in class k. Clearly, minimizing the above empirical loss function will lead to the minimization of the classification error. As a result, Eq. (24) is called the MCE criterion [10, 11, 20]. The class parameter set Λ is therefore obtained by minimizing the loss function through the steepest gradient descent algorithm. This is an interative algorithm and the iteration rules are:

$$\Lambda_{t+1} = \Lambda_t - \epsilon \nabla L(\Lambda)|_{\Lambda = \Lambda_t} \tag{25}$$

$$\nabla L(\Lambda) = \begin{bmatrix} \partial L / \partial \lambda_1 \\ \vdots \\ \partial L / \partial \lambda_d \end{bmatrix} \tag{26}$$

where t denotes tth iteration, $\lambda_1, \ldots, \lambda_d \in \Lambda$ are parameters, $\epsilon > 0$ is the adaption constant. For $s = 1, 2, \ldots, d$, the gradient $\nabla L(\Lambda)$ can be computed as follows:

$$\frac{\partial L}{\partial \lambda_s} = \xi \sum_{i=1}^{N_k} L^{(i)}\big(1 - L^{(i)}\big) \frac{\partial_{g_k}\big(x^{(i)}, \Lambda\big)}{\partial \lambda_s}, \quad \text{if } \lambda_s \in \textit{class } k \tag{27}$$

$$\frac{\partial L}{\partial \lambda_s} = -\xi \sum_{i=1}^{N_j} L^{(i)}\big(1 - L^{(i)}\big) \frac{\partial_{g_j}\big(x^{(i)}, \Lambda\big)}{\partial \lambda_s}, \quad \text{if } \lambda_s \in \textit{class } j \tag{28}$$

3.2. Using MCE Training Algorithms for Dimensionality Reduction

As with other feature extraction methods, MCE reduces feature dimensionality by projecting the input vector into a lower dimensional space by a linear transformation $T_{m \times p}$, where $m < p$,

$$y = Tx \tag{29}$$

The loss function is then defined on this m-dimensional space using the transformed vector y, rather than x, we get:

$$L(\tilde{\Lambda}, T) = E\{l(d_k(y, \tilde{\Lambda}))\} = E\{l(d_k(Tx\tilde{\Lambda}))\} \tag{30}$$

where $\tilde{\Lambda}$ is the parameter set obtained in the reduced m-dimensional space. Since Eq. (30) is a function of T, the elements in T are optimized together with the parameter set $\tilde{\Lambda}$ in the same gradient descent procedure. The adaption rule for T is:

$$T_{nq}(t+1) = T_{nq}(t) - \epsilon \left.\frac{\partial L}{\partial T_{nq}}\right|_{T_{nq} = T_{nq}(t)} \tag{31}$$

and

$$\frac{\partial L}{\partial T_{nq}} = \xi \sum_{k=1}^{K} \sum_{i=1}^{N_k} L^{(i)}\big(1 - L^{(i)}\big) \times \left(\frac{\partial_{g_k}\big(Tx^{(i)}, \tilde{\Lambda}\big)}{\partial T_{nq}} - \frac{\partial_{g_j}\big(Tx^{(i)}, \tilde{\Lambda}\big)}{\partial T_{nq}} \right) \tag{32}$$

where t denotes tth iteration, ϵ is the adaption constant and n and q are the row and column indicators. It is worth noting that in this case the parameter set optimized in MCE training procedure is $\tilde{\Lambda}$ in Eq. (30), rather than Λ obtained from the original p-dimensional space.

3.3. An Alternative MCE Training Algorithm

The purpose of defining the misclassification measure is to obtain the largest discrimination between $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$. Therefore the control of the joint behavior of $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$ is essential to a successful MCE training. The conventional definitions in Eqs. (21) and (22) combine $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$ linearly. Linear combinations, however, have a very loose control of the joint behavior of $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$. To enhance MCE's ability to control the joint behavior of discriminant functions, we propose an alternative definition of misclassification measure which combines $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$ non-linearly. The alternative definition also comes from Bayes decision rule. We first rewrite Eq. (19) as follows:

$$x \in \textit{Class } k \quad \text{if } \frac{\max_{\text{for all } i \neq k} g_i(x, \Lambda)}{gk(x, \Lambda)} < 1 \tag{33}$$

Then in the reduced m-dimensional subspace, we define the misclassification measure $d_k(x, \tilde{\Lambda}, T)$ as an approximate of the L.H.S of Eq. (33):

$$d_k(x, \tilde{\Lambda}, T) = \frac{\left[\frac{1}{N-1} \sum_{\text{for all } i \neq k} g_i(Tx, \tilde{\Lambda})^{\eta}\right]^{1/\eta}}{g_k(Tx, \tilde{\Lambda})} \tag{34}$$

To the extreme case, i.e. $\eta \to \infty$, Eq. (34) becomes:

$$d_k(x, \tilde{\Lambda}, T) = \frac{g_j(Tx, \tilde{\Lambda})}{g_k(Tx, \tilde{\Lambda})} \tag{35}$$

The class parameters and transformation matrix are optimized using the same adaption rules as shown in Eqs. (25) and (31). The gradients of $\tilde{\Lambda}$ and T are calculated as follows:

$$\frac{\partial L}{\partial \tilde{\lambda}_s} = -\xi \sum_{i=1}^{N_j} L^{(i)}\big(1 - L^{(i)}\big) \times \frac{g_j\big(Tx^{(i)}, \tilde{\Lambda}\big)}{\big[g_k\big(Tx^{(i)}, \tilde{\Lambda}\big)\big]^2} \frac{\partial_{g_k}\big(Tx^{(i)}, \tilde{\Lambda}\big)}{\partial \tilde{\lambda}_s}, \quad \text{if } \tilde{\lambda}_s \in \textit{class } k \tag{36}$$

$$\frac{\partial L}{\partial \tilde{\lambda}_s} = \xi \sum_{i=1}^{N_k} L^{(i)}\left(1 - L^{(i)}\right) \frac{1}{g_k\left(Tx^{(i)}, \tilde{\Lambda}\right)} \frac{\partial_{g_j}\left(Tx^{(i)}, \tilde{\Lambda}\right)}{\partial \tilde{\lambda}_s}, \quad \text{if } \tilde{\lambda}_i \in \mathit{class}\ j \tag{37}$$

and

$$\frac{\partial L}{\partial T_{nq}} = \xi \sum_{k=1}^{K} \sum_{i=1}^{N_k} L^{(i)}\left(1 - L^{(i)}\right) \times \frac{\frac{\partial_{g_j}(Tx^{(i)}, \tilde{\Lambda})}{\partial T_{nq}} g_k\left(Tx^{(i)}, \tilde{\Lambda}\right) - \frac{\partial_{g_k}(Tx^{(i)}, \tilde{\Lambda})}{\partial T_{nq}} g_j\left(Tx^{(i)}, \tilde{\Lambda}\right)}{\left[g_k\left(Tx^{(i)}, \tilde{\Lambda}\right)\right]^2} \tag{38}$$

where $\tilde{\lambda}_s \in \tilde{\Lambda}$, $s = 1, \ldots, d$, n and q are the row and column indicators of transformation matrix T.

3.4. A Comparison of Two Forms of MCE Training Algorithms

The proposed alternative form of MCE training algorithm differs from the conventional one in that the misclassification measure is a non-linear combination of discriminant functions. To compare these two forms of MCE training algorithms, we use a $g_k(x, \Lambda)$-$g_j(x, \Lambda)$ decision plane to show their behaviors in the training process. The vertical axis of the decision plane is $g_k(x, \Lambda)$, which represents the discriminant of a vector x to its desired class k. The horizontal axis is $g_j(x, \Lambda)$, representing the largest discriminant of x among all the classes other than k. The decision line is $g_k(x, \Lambda) = g_j(x, \Lambda)$. Driven by the training algorithms, all the training data move in this plane throughout the training process. The behaviors of the training algorithms are therefore demonstrated by the tracks of data. If the data move towards the top left of the decision plane, both the absolute and relative differences between $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$ increase. Therefore the training process is effective and robust. If, however, the data move towards the top right of the plane, the relative difference between $g_k(x, \Lambda)$ and $g_j(x, \Lambda)$ does not increase significantly despite an increase in the absolute difference. Furthermore, the training can become divergent if not precisely controlled. In this case the training process is not desirable. Figure 1(a) shows the theoretical behaviors of these two forms of MCE in the training process and Fig. 1(b) shows their real behavior. The data used in Fig. 1(b) is randomly selected from Deterding database. These two figures show that the proposed alternative MCE training algorithm is more robust than the conventional one.

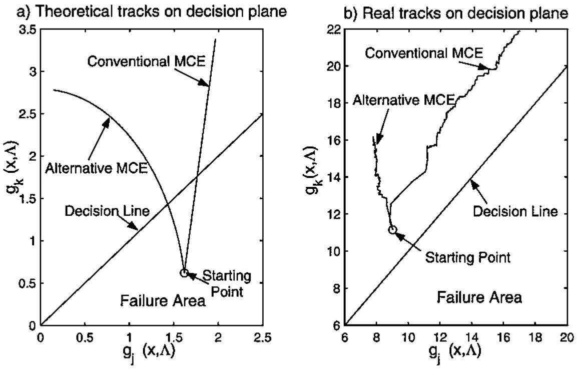

Figure 1. Theoretical and real tracks of a data moving in the decision plane.

4. Classification Experiments

4.1. Databases

An evaluation of these feature dimensionality reduction algorithms was made on two different databases. The first one is Deterding vowel database, which has 11 vowel classes as shown in the Table 1.

Each of these 11 vowels are uttered 6 times by 15 different speakers. This gives a total of 990 vowel tokens. A central frame of speech signal is excised from each of these 990 vowel tokens. A 10th order linear prediction analysis is carried out for each frame resulting in 10 log-area parameters. These 10 parameters defines the original 10 dimensional feature space. 528 frames from the eight speakers are used to train the models and 462 frames from the seven speakers are used to test the models.

The other database is D. German's GLASS database which contains the measurements of the chemical constitutions in terms of their oxide content (Na, Mg, Al, Si, K, Ca, Ba and Fe) and the refractive index of the glass, manufactured through two different processes. The database has 163 instances, of which 87 measurements are made on glass manufactured through the float process and 76 on glass through non-float process. Each measurement has 10 numeric-valued attributes.

Table 1. Vowels and words used in Deterding database.

Vowel	Word	Vowel	Word	Vowel	Word	Vowel	Word
i	heed	O	hod	I	hid	C:	hoard
E	head	U	hood	A	had	u:	who'd
a:	hard	3:	heard	Y	hud		

The reason for using these two databases is that they have been studied by other researchers [1, 12, 21], so it is easy to compare the results. For the sake of convenience, we denote conventional MCE as MCE(con) and the alternative MCE as MCE(alt) in the figures.

4.2. Classification Results

Four algorithms, MCE(con), MCE(alt), LDA and PCA, were used in the classification experiments. All of them used the Mahalanobis distance classifier in the classification tasks. Figure 2 shows the results when using the Deterding database in different dimensional spaces. We can make the following observations:

- During the training process, all of the four algorithms perform better when the training is carried out in the original feature space. But this is not the case in the testing process where the best results are usually obtained when the dimensionality of the feature space has been reduced to half.
- During the training process, recognition rates increase with dimensionality. During the testing process, however, no regular pattern of changes in recognition rates with dimensionality can be observed.
- The overall performance of the alternative MCE training algorithm is better than the other three algorithms on both training data and testing data.

Table 2 shows the results of the four algorithms obtained from GLASS dataset. The results of conventional MCE and alternative MCE are given in Columns 2 and 3 respectively and the results of LDA and PCA are in Columns 4 and 5, respectively. Similar observations can be made from the table, except that (1) the performances of two MCE training algorithms are much better than those of LDA and PCA; (2) the recognition rate of LDA decreases when the dimension of the features increases. In comparison, the highest classification rate achieved in [1] is 69.3% by means of Mahalanobis Linear Transformation training.

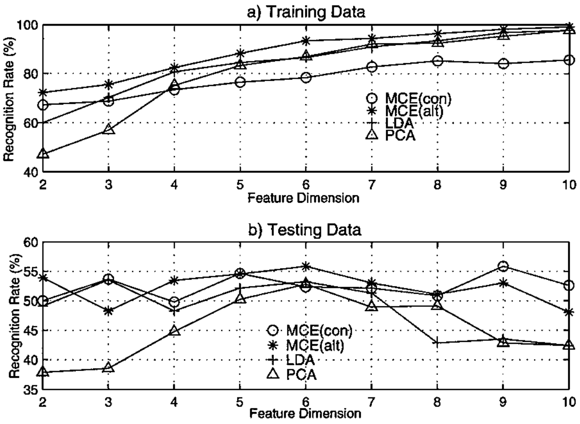

Figure 2. Comparison of the recognition rates of MCE(con), MCE(alt), LDA and PCA on Daterding database.

Table 2. Results on GLASS data (in %).

DIM	MCE(con)	MCE(alt)	LDA	PCA
2	77.9	77.9	68.1	48.5
3	75.5	83.4	64.4	49.1
4	79.1	80.4	63.2	60.7
5	77.3	80.4	65.0	63.2
6	79.7	82.8	62.0	63.8
7	76.7	83.4	63.2	61.4

4.3. Initialization of the Transformation Matrix

One of the major concerns about MCE training for dimensionality reduction is the initialization of the transformation matrix. This is because the gradient descent method used in MCE does not guarantee the global minimum value. It is largely dependent on the choice of the starting point, as shown in Fig. 3. Since in MCE training, the starting point is directly determined by the initial transformation matrix, the initialization of the transformation matrix is of high importance. In MCE training, a unit matrix is usually chosen as the initial matrix for the transformation. This is equivalent to using only the first m features in a feature vector to start the training. This type of initialization is poor because

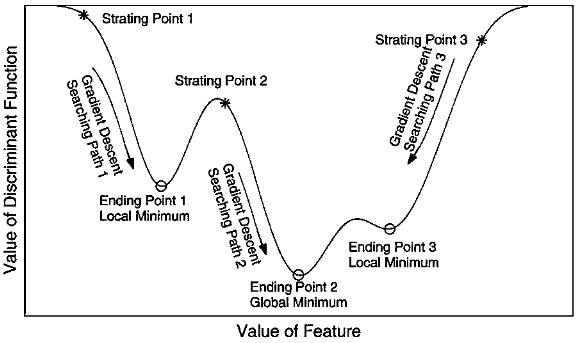
Strating Point 1
Gradient Descent Searching Path 1
Ending Point 1 Local Minimum
Strating Point 2
Gradient Descent Searching Path 2
Ending Point 2 Global Minimum
Strating Point 3
Gradient Descent Searching Path 3
Ending Point 3 Local Minimum
Value of Discriminant Function
Value of Feature

Figure 3. Effects of the choices of the starting point on MCE training.

the first m features do not always store the most useful information. Hence, such initialization does not necessarily make the training process approach the global minimum. From searching point of view, we can regard the initialization of transformation matrix and MCE training as two sequential search procedures: one is general but rough and the other, local but thorough. The former procedure will provide a global optimized starting point and the latter will make a thorough search to find the relevant local minimum. At present, no suitable criteria exist for general search. In this paper, we employed LDA and PCA for the general search. The results are compared to those of MCE training that begins with unit matrix.

Initialization of the transformation matrix is one of the key procedures in MCE training. In this paper we explored the use of LDA and PCA as the pre-processing procedure to provide initial matrix for transformation. The alternative MCE training algorithm initialized with LDA and PCA are denoted as MCE(alt) + LDA and MCE(alt) + PCA, respectively. The MCE training initialized with unit transformation matrix is denoted as MCE(alt) + UNIT.

The results for Deterding database are shown in Figs. 4 and 5, while for the GLASS data, in Table 3.

Observations from these results can be summarized as follows:

- Performances of MCE training algorithm on both Deterding and GLASS databases improve when LDA is used to initialize the transformation matrix.
- The performance of MCE training algorithm on GLASS database improves when PCA is used to initialize the transformation matrix, while on the Deterding database, remains unchanged.
- MCE(alt) + LDA demonstrates the best performance among MCE(alt) + UNIT, MCE(alt) + PCA, LDA and PCA except for dimension 5 on the Deterding database and dimensions 2, 3 and 4 on the GLASS database.
- Performances of MCE(alt) + LDA and MCE(alt) + PCA on testing data of Deterding database show that the best classification results are usually obtained when the dimensionality is reduced to 50%–70%.

Table 3. Results on GLASS data (in %).

DIM	MCE(alt) + UNIT	MCE(alt) + LDA	MCE(alt) + PCA
2	77.9	81.0	82.8
3	83.4	82.2	84.0
4	80.4	82.8	83.4
5	80.4	84.7	82.8
6	82.8	84.1	82.2
7	83.4	82.8	82.8

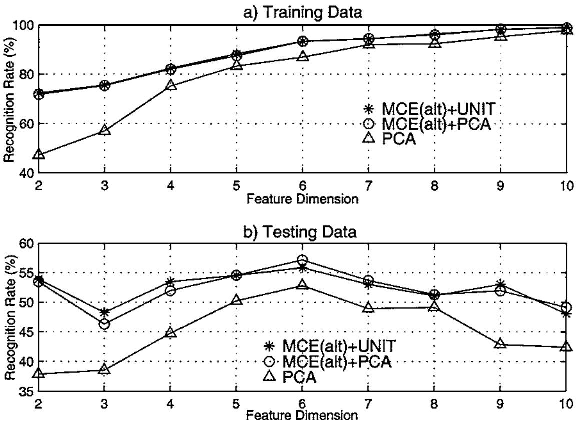

Figure 5. Comparison of the recognition rates of MCE(alt) + UNIT, MCE(alt) + PCA, PCA on Deterding database.

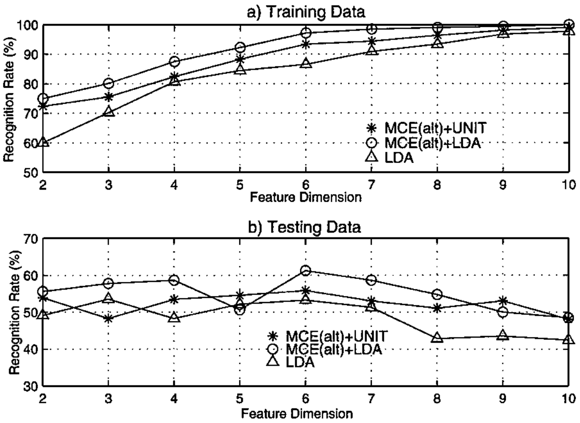

Figure 4. Comparison of the recognition rates of MCE(alt) + UNIT, MCE(alt) + LDA, LDA on Deterding database.

5. Conclusion

The MCE training algorithms, especially the proposed alternative MCE training algorithm, usually provides better results than those by LDA and PCA when used for dimensionality reduction tasks. To further enhance the performance of the proposed MCE training algorithm, a general pre-search procedure was used to initialize the transformation matrix. In this paper, we explored several criteria and LDA yields the best results. However, LDA is still far from being perfectly suited to this procedure and further research is required.

References

1. H. Brunzell and J. Eriksson, “Feature Reduction for Classification of Multidimensional Data,” *Pattern Recognition*, vol. 33, 2000, pp. 1741–1748.
2. K.K. Paliwal, “Dimensionality Reduction of the Enhanced Feature Set for the HMM-Based Speech Recognizer,” *Digital Signal Processing*, no. 2, 1992, pp. 157–173.
3. E.L. Bocchieri and J.G. Wilpon, “Discriminative Analysis for Feature Reduction in Automatic Speech Recognition,” in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing*, vol. 1, 1992, pp. 501–504.
4. C.J. Leggetter, “Improved Acoustic Modelling for HMMs Using Linear Transformations,” Ph.D. Thesis, University of Cambridge, 1995.
5. I.T. Jolliffe, *Principal Component Analysis*, New York: Springer-Verlag, 1986.
6. W.J. Krzanowski, “Principal Component Analysis in the Presence of Group Structure,” *Applied Statistics*, vol. 33, 1984, pp. 164–168.
7. M. Hotelling, “Analysis of a Complex of Statistical Variables into Principal Components,” *Journal of Educational Psychology*, vol. 24, 1933, pp. 498–520.
8. D.X. Sun, “Feature Dimension Reduction Using Reduced-Rank Maximum Likelihood Estimation for Hidden Markov Model,” in *Proceedings of Internation Conference on Spoken Language Processing*, Philadelphia, USA, 1996, pp. 244–247.
9. W.L. Poston and D.J. Marchette, “Recursive Dimensionality Reduction Using Fisher’s Linear Discriminant,” *Pattern Recognition*, vol. 31, no. 7, 1998, pp. 881–888.
10. S. Katagiri, C.H. Lee, and B.H. Juang, “A Generalized Probabilistic Descent Method,” in *Proceedings of the Acoustic Society of Japan*, Fall Meeting, 1990, pp. 141–142.
11. B.H. Juang and S. Katagiri, “Discriminative Learning for Minimum Error Classification,” *IEEE Transactions on Signal Processing*, vol. 40, no. 12, 1992.
12. K.K. Paliwal, M. Bacchiani, and Y. Sagisaka, “Simultaneous Design of Feature Extractor and Pattern Classifier Using the Minimum Classification Error Training Algorithm,” in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, Boston, USA, September 1995, pp. 67–76.
13. E. McDermott and S. Katagiri, “Prototype-Based Minimum Classification Error/Generalized Probabilistic Descent Training for Various Speech Units,” *Computer Speech and Language*, vol. 8, no. 8, 1994, pp. 351–368.
14. P.C. Chang, S.H. Chen, and B.H. Juang, “Discriminative Analysis of Distortion Sequences in Speech Recognition,” in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing*, vol. 1, 1991, pp. 549–552.
15. I. Komori and S. Katagiri, “GPD Training of Dynamic Programming-Based Speech Recognizer,” *Journal of Acoustical Society of Japan (E)*, vol. 13, no. 6, 1992, pp. 341–349.
16. W. Chou, B.H. Juang, and C.H. Lee, “Segmental GPD Training of HMM Based Speech Recognizer,” in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing*, 1992, vol. 1, pp. 473–476.
17. D. Rainton and S. Sagayama, “Minimum Error Classification Training of HMMs-Implementation Details and Experimental Results,” *Journal of Acoustical Society of Japan (E)*, vol. 13, no. 6, 1992, pp. 379–387.
18. C.S. Liu, C.H. Lee, W. Chou, and B.H. Juang, “A Study on Minimum Error Discriminative Training for Speaker Recognition,” *Journal of Acoustical Society of America*, vol. 97, no. 1, 1995, pp. 637–648.
19. N. Kumar and A.G. Andreou, “A Generalization of Linear Discriminant Analysis in Maximum Likelihood Framework,” in *Proceedings of the Joint Statistical Meeting*, Statistical Computing Section, Chichago, Aug. 4–8, 1996.
20. B. Tian and M.R. Azimi-Sadjadi, “Comparison of Two Different PNN Training Approaches for Satellite Cloud Data Classification,” *IEEE Transactions on Neural Networks*, vol. 12, no. 1, 2001, pp. 164–168.
21. S. Aeberhard, O. de Vel, and D. Coomans, “Comparative Analysis of Statistical Pattern Recognition Methods in High Dimensional Settings,” *Pattern Recognition*, vol. 27, no. 8, 1994, pp. 1065–1077.
22. K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis*, New York: Academic Press/Harcourt Brace, 1979.
23. A. Biem and S. Katagiri, “Feature Extraction Based on Minimum Classification Error/Generalized Probabilistic Descent Method,” in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing*, vol. 2, 1993, pp. 275–278.
24. P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Englewood Cliffs, NJ: Prentice-Hall, 1982.
25. K. Fukunaga and D.R. Olsen, “An Algorithm for Finding Intrinsic Dimensionality of Data,” *IEEE Transactions on Computers*, vol. C-20, no. 2, 1971, pp. 176–183.
26. G.W. Cottrell, “Principal Componants Analysis of Images via Back Propagation,” *SPIE Proceedings in Visual Communication and Image Processing*, vol. 1001, 1988, pp. 1070–1077.
27. T.K. Leen, “Dynamics of Learning in Linear Feature-Discovery Networks,” *Network: Computation in Neural Systems*, vol. 2, 1991, pp. 85–105.
28. J. Yang and G.A. Dumont, “Classification of Acoustical Emmission Signals via Hebbian Feature Extraction,” *IEEE Proceedings of the IJCNN*, Piscataway, NJ, 1991, vol. 1, pp. 113–118.
29. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, New York: John Wiley, 1973.
30. X. Wang and K.K. Paliwal, “Using Minimum Classification Error Training in Dimensionality Reduction,” in *Proceedings of the 2000 IEEE Workshop on Neural Networks for Signal Processing X*, 2000, pp. 338–345.

Xuechuan Wang was born in China in May 1971. He received his B.E. degree in Mechanical Engineering from Hunan University,

Changsha, China in 1992 and M.E. degree in Signal Processing from Huazhong University of Science and Technology, Wuhan, China in 1995. From 1996 to 1998, he worked at the same university. He has been a Ph.D. candidate since 1998 in Signal Processing Lab at Griffith University, Brisbane, Australia. His current research interests include discriminant analysis, features extraction and speech recognition.
wang@me.gu.edu.au

Kuldip K. PALIWAL received the B.S. degree from Agra University, India in 1969, the M.S. degree from Aligarh University, India, in 1971 and the Ph.D. degree from Bombay University, India, in 1978. Since 1993, he has been a Professor (Chair, Communication/Information Engineering) at the Griffith University, Brisbane, Australia. He has worked at a number of organizations including Tata Institute of Fundamental Research, Bombay, India, Norwegian Institute of Technology, Trondheim, Norway, University of Keele, U.K., AT&T Bell Laboratories, Murray Hill, New Jersey, U.S.A. and Advanced Telecommunication Research (ATR) Laboratories, Kyoto, Japan. He has co-edited two books: *Speech Coding and Synthesis* (Elsevier, 1995) and *Speech and Speaker Recognition: Advanced Topics* (Kluwer, 1996). He has published more than 100 papers in international journals. He is a recepient of the 1995 IEEE Signal Processing Society Senior Award. He has been an Associate Editor of the IEEE Transactions on Speech and Audio Processing, and IEEE Signal Processing Letters. His current research interests include speech processing, image coding and neural networks.
K.Paliwal@me.gu.edu.au