Generalized MCE Training Algorithm for Feature Dimensionality Reduction

X. Wang and K. K. Paliwal

Abstract— **Dimensionality reduction is an important problem in pattern recognition. Reducing the dimensionality of feature can improve the effecitveness and efficiency of pattern recognition algorithms. Minimum Classification Error(MCE) training algorithm is a power tool for dimensionality resuction. However, MCE training process is a type of thorough search process for the local minimum, global minimum can not be guaranteed by this process. In this paper, a generalized MCE training algorithm is proposed. This algorithm uses a general search process for searching the generalized starting point. Then conventional MCE training algorithm is used to search for the minimum.**

I. 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 [8] 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. However, the recognition results obtained using the feature selection methods depend heavily on the ranking measure used [2]. 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. LDA pursues the largest ratio of between-class variation and within-class variation when projecting the original feature to a subspace.

MCE training algorithm is a type of discriminant training algorithm. It is proposed to mend the shortcomings of traditional discriminant training [2], [4]. As pointed out by Juang and Katagiri [3], 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. 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 and HMM based speech and speaker recognition.

II. MCE Training Algorithm

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

$$x \in Class\ k \quad if \quad g_k(x,\Lambda) = \max_{for\ all\ i \in K} g_i(x,\Lambda) \tag{1}$$

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. We define the misclassification measure as

$$d_k(x,\Lambda) = -g_k(x,\Lambda) + [\frac{1}{N-1} \sum_{for\ all\ i \neq k} (g_i(x,\Lambda))^{\eta}]^{1/\eta}, \tag{2}$$

where η is a positive number, $g_k(x,\Lambda)$ is the discriminant function of observation x and T is a $m \times p, m < p$ transformation matrix to project input vectors into a lower dimensional space. When η approaches ∞, it reduces to

$$d_k(x,\Lambda) = -g_k(x,\Lambda) + g_j(x,\Lambda), \tag{3}$$

where class j has the largest discriminant value among all the classes other than class k. Clearly, $d_k(x,\Lambda) > 0$ implies misclassification, $d_k(x,\Lambda) < 0$ means correct classification and $d_k(x,\Lambda) = 0$ suggests that x sits on the boundary. Since MCE training algorithm uses gradient descent method for searching the minimum, the objective function, i.e. the misclassification measure defined in Eq.(3), is linear and thus not ideal for the gradient descent searching. In this paper we give an non-linear alternative form of misclassification measure, which uses ratio between discrimination functions. The alternative form is given by:

$$d_k(x,\Lambda) = \frac{[\frac{1}{N-1} \sum_{for\ all\ i \neq k} g_i(x,\Lambda)^{\eta}]^{1/\eta}}{g_k(x,\Lambda)} \tag{4}$$

To the extreme case, i.e. $\eta \to \infty$, Eqn. (4) becomes:

$$d_k(x,\Lambda) = \frac{g_j(x,\Lambda)}{g_k(x,\Lambda)} \tag{5}$$

The decision plane decided by Eq.(4) and Eq.(5) is thus $d_k(x,\Lambda) = 1$. The loss function is defined as a monotonic function of misclassification measure:

$$l_k(x,\Lambda) = f(d_k(x,\Lambda)) = \frac{1}{1 + e^{-\xi d_k(x,\Lambda)}} \tag{6}$$

where $\xi > 0$. For a training set $\mathcal{X}$. For a training set $\mathcal{X}$, the empirical loss is defined as:

$$L(\Lambda) = E\{l_k(x,\Lambda)\} = \sum_{k=1}^{K}\sum_{i=1}^{N_k} l_k(x^{(i)},\Lambda) \tag{7}$$

where N_k is the number of samples in class k. The class parameter set Λ is optimized by minimizing the loss function through the steepest gradient descent algorithm. The iteration rules are:

$$\Lambda_{t+1} = \Lambda_t - \varepsilon \nabla L(\Lambda)|_{\Lambda=\Lambda_t} \tag{8}$$

where t denotes tth iteration, $\lambda_1, \cdots, \lambda_d \in \Lambda$ are parameters, $\varepsilon > 0$ is the adaption constant.

III. MCE for Dimensionality Reduction

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

$$y = Tx \tag{9}$$

The parameter set $\tilde{\Lambda}$ is therefore set up and trained in this m-dimensional space. Accordingly, misclassification measure is reformulated over $\tilde{\Lambda}$ and uses the transformed vector y as input. For convience, we will use Tx rather than y in the following. The misclassification measure forconventional MCE is redefined as:

$$d_k(Tx,\tilde{\Lambda}) = -g_k(Tx,\tilde{\Lambda}) + [\frac{1}{N-1}\sum_{for all\, i\neq k}(g_i(Tx,\tilde{\Lambda}))^{\eta}]^{1/\eta} \tag{10}$$

When η approaches ∞, the misclassification measure becomes:

$$d_k(Tx,\tilde{\Lambda}) = -g_k(Tx,\tilde{\Lambda}) + g_j(Tx,\tilde{\Lambda}) \tag{11}$$

For alternative MCE, the misclassification measure is redefined as:

$$d_k(Tx,\tilde{\Lambda}) = \frac{[\frac{1}{N-1}\sum_{for all\, i\neq k} g_i(Tx,\tilde{\Lambda})^{\eta}]^{1/\eta}}{g_k(Tx,\tilde{\Lambda})} \tag{12}$$

To the extreme case, i.e. $\eta \to \infty$, Eqn. (12) becomes:

$$d_k(Tx,\tilde{\Lambda}) = \frac{g_j(Tx,\tilde{\Lambda})}{g_k(Tx,\tilde{\Lambda})} \tag{13}$$

The loss of classifying an observation vector x is then calculated via its transformed vector y:

$$l(x,\tilde{\Lambda},T) = l(d_k(Tx,\tilde{\Lambda})) = \frac{1}{1+e^{-\alpha d(Tx,\tilde{\Lambda})}} \tag{14}$$

The total loss over the whole observation set is given by:

$$L(\tilde{\Lambda},T) = E\{l(d_k(Tx,\tilde{\Lambda}))\} \tag{15}$$

Since Eqn. (15) is a function of T, the elements in T can be optimized together with the parameter set $\tilde{\Lambda}$ in the same gradient descent procedure. The adaption rule for T is:

$$T_{sq}(t+1) = T_{sq}(t) - \varepsilon \frac{\partial L}{\partial T_{sq}}\bigg|_{T_{sq}=T_{sq}(t)} \tag{16}$$

where t denotes tth iteration, ε is the adaption constant or learning rate and s and q are the row and column indicators of transformation matrix T. For conventional MCE,

$$l\frac{\partial L}{\partial T_{sq}} = \xi\sum_{k=1}^{K}\sum_{i=1}^{N_k} L^{(i)}(1-L^{(i)})\left(\frac{\partial g_k(Tx^{(i)},\tilde{\Lambda})}{\partial T_{sq}} - \frac{\partial g_j(Tx^{(i)},\tilde{\Lambda})}{\partial T_{sq}}\right) \tag{17}$$

and for alternative MCE,

$$\frac{\partial L}{\partial T_{sq}} = \xi\sum_{k=1}^{K}\sum_{i=1}^{N_k} L^{(i)}(1-L^{(i)}) \frac{\frac{\partial g_j(Tx^{(i)},\tilde{\Lambda})}{\partial T_{sq}} g_k(Tx^{(i)},\tilde{\Lambda}) - \frac{\partial g_k(Tx^{(i)},\tilde{\Lambda})}{\partial T_{sq}} g_j(Tx^{(i)},\tilde{\Lambda})}{[g_k(Tx^{(i)},\tilde{\Lambda})]^2} \tag{18}$$

IV. Generalized MCE Training Algorithm

Since MCE training algorithm uses gradient descent method for searching, the training process is a type of thorough search process for the local minimum. Global minimum can not be guaranteed by this process. The optimality of MCE training process largely depends on the initialization of the parameter set $\{\mu,\Sigma,T\}$. Among these parameters, transformation matrix T is crucial to the success of MCE training process, because T decides both what kind of and how much class information would be brought into the training subspace. Figure 1 shows the importance of transformation matrix to the training process. [5] gives a popular way to initialize the training process, in which the transformation matrix T is taken to be a unity matrix. The prototype vectors for different classes are initialized by their maximum likelihood estimates (i.e. by their conditioned means and/or variances). However, in many cases, this is a convenient way of initialization rather than an effective way because the classification criterion has not been considered in

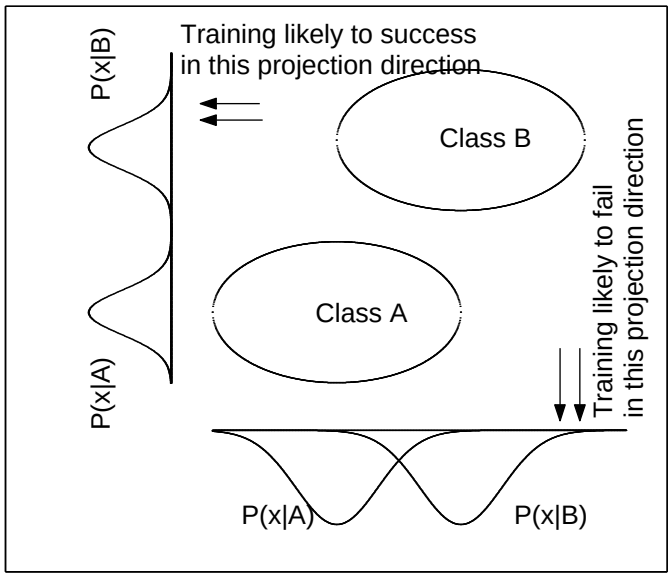

Fig. 1. Importance on choosing transformation matrix, which decides the projection direction.

this type of initialization. In order to increase the generalization of MCE training algorithm, it is necessary to embed the classification criteria into the initialization process. From searching point of view, we can regard 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. Figure 2 compares the normal MCE training process to generalized MCE training process. So far, no criterion on general search process has been proposed. However we can employ current feature extraction methods into this process. Two criteria can be the suitable choices. One is PCA, which decorrelates the features by projecting them along the principal component directions. The most common derivation of PCA is in terms of a standardised linear projection which maximises the variance in the projected space [6]. For a given p-dimensional data set $\mathcal{X}$, the m principal axes $T_1, T_2, \cdots, T_m$, where $1 \leq m \leq p$, are orthonomal axes onto which the retained variance is maximum in the projected space. Generally, $T_1, T_2, \cdots, T_m$ 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 $x_i \in \mathcal{X}$, μ is the sample mean and N is the number of samples, so that:

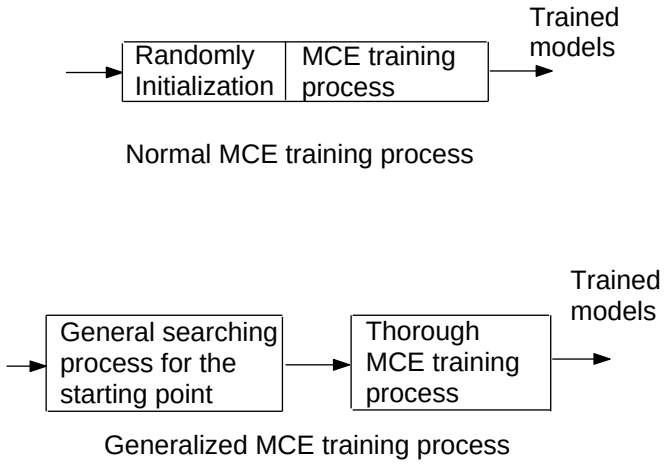

Fig. 2. Comparison between normal MCE training process and generalized MCE training process.

$$ST_i = \lambda_i T_i \quad i \in 1, \cdots, m \tag{19}$$

where λ_i is the ith largest eigenvalue of S. The transformation can then be given by combining $T_1, T_2, \cdots, T_m$, which yields:

$$T = [T_1, \cdots, T_m] \tag{20}$$

Another criterion is LDA, which pursues the largest ratio of between-class variation and within-class variation when projecting the original feature to a subspace [7]. Linear discriminant is defined as the linear functions $T^T x$ for which the criterion function

$$J(T) = \frac{|S_B|}{|S_B|} = \frac{T^T S_B T}{T^T S_W T} \tag{21}$$

is maximum, where S_W is within-class covariance matrix and given by:

$$S_W = \sum_{j=1}^{K} \frac{1}{N_j} \sum_{i=1}^{N_j} (x_{ji} - \mu_j)(x_{ji} - \mu_j)^T \tag{22}$$

S_B is between-class covariance matrix and defined as:

$$S_B = \frac{1}{N} \sum_{j=1}^{K} N_j (\mu_j - \mu)(\mu_j - \mu)^T \tag{23}$$

where $N = \sum_{j=1}^{K} N_j$, μ_j is the individual class mean and μ is the global mean. It can be shown that the solution of Eq. (21) for T is in fact the matrix of the leading m eigenvectors of $S_W^{-1} S_B$.

V. Experiments Results

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.

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		

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.

The reason for using these two databases is that they have been studied by other researchers[5], [8], [9], so it is easy to compare the results. In this paper, the alternative MCE defined in Eq. (5) is used. For the sake of convenience, we denote it as MCE(alt) in the figures. The Generalized MCE using LDA for general search is denoted as MCE+LDA, the one using PCA is denoted as MCE+PCA. The results are compared to those of normal MCE training that begins with unit matrix. The results are also compare to those of LDA and PCA, which are popular methods for feature dimensionality reduction.

The results for Deterding database are shown in Figures 3 and 4, while for the GLASS data, in Table 2.

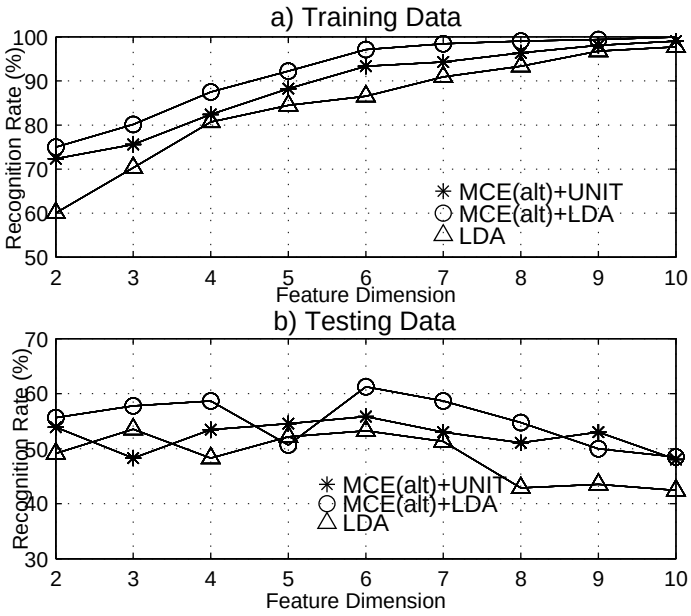

Fig. 3. Comparison of the recognition rates of MCE, MCE+LDA, LDA on Deterding database.

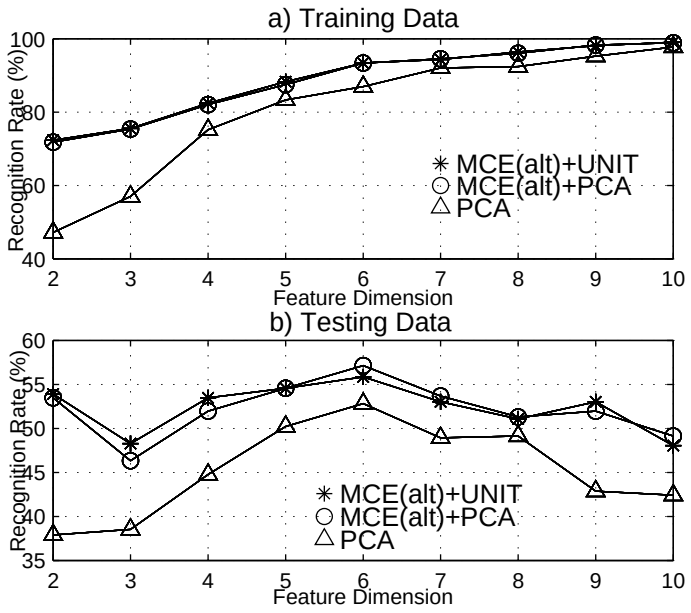

Fig. 4. Comparison of the recognition rates of MCE, MCE+PCA, PCA on Deterding database.

Table 2 Results on GLASS data (in %)

DIM	MCE	MCE+LDA	MCE+PCA	LDA	PCA
2	77.9	81.0	82.8	68.1	48.5
3	80.4	82.2	84.0	64.4	49.1
4	80.4	82.8	83.4	63.2	60.7
5	80.4	84.7	82.8	65.0	63.2
6	81.8	84.1	82.2	62.0	63.8
7	81.4	84.8	82.8	63.2	61.4

Observations can be summarized as follows:

- MCE training algorithms perform better than LDA and PCA in all the feature dimensionality reduction experiments. The advantages of MCE training algorithms are especially significant on Glass data.
- When using LDA for general search, generalized MCE training algorithm performs better than normal MCE training algorithm on both Deterding and GLASS databases.
- When using PCA for general search, generalized MCE training algorithm has nearly the same performance as normal MCE training algorithm on Deterding data. Its performance on GLASS databases is better than that of MCE training algorithm.

VI. Conclusion

The results show that MCE training algorithm is suitable for feature dimensionality reduction and performs better than LDA and PCA. The performance of MCE training algorithm can be further improved after generalization implementation. Generalized MCE training algorithms can achieve better training results than normal MCE training algorithm. However it largely depends on the criterion used for general searching process. Although there are still lacks of effective criterion to supervise the gerenal searching process, current feature extraction methods can be easily employed in this process. Among these methods, LDA shows its advantage over other methods in generalized MCE training process.

References

[1] H.Brunzell and J.Eriksson, “feature Reduction for Classification of Multidimensional Data”, *Pattern Recognition*, 33, pp. 1741-1748, 2000

[2] K.K.Paliwal, “Dimensionality Reduction of the Enhanced Feature Set for the HMM-Based Speech Recognizer”, *Digital Signal Processing*, No. 2, pp. 157-173, 1992

[3] S.Katagiri, C.H.Lee and B.H.Juang, “A Generalized Probabilistic Descent Method”, *Proceedings of the Acoustic Society of Japan, Fall Meeting*, pp. 141-142, 1990

[4] B.H.Juang and S.Katagiri, “Discriminative Learning for Minimum Error Classification”,*IEEE Transactions on Signal Processing*, Vol. 40, No. 12, December, 1992

[5] K.K.Paliwal, M.Bacchiani and Y.Sagisaka, “Simultaneous Design of Feature Extractor and Pattern Classifier Using the Minimum Classification Error Training Algorithm”, *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, Boston, USA, pp. 67-76, September, 1995

[6] M.Hotelling, “Analysis of a Complex of Statistical Variables into Principal Components”, *Journal of Educational Psychology*, 24, pp. 498-520, 1933

[7] D.X.Sun, “Feature Dimension Reduction Using Reduced-Rank Maximum Likelihood Estimation For Hidden Markov Model”, *Proceedings of Internation Conference on Spoken Language Processing*, Philadelphia, USA, pp. 244-247, 1996

[8] H.Brunzell and J.Eriksson, “feature Reduction for Classification of Multidimensional Data”, *Pattern Recognition*, 33, pp. 1741-1748, 2000

[9] S.Aeberhard, O. de Vel and D.Coomans, “Comparative Analysis of Statistical Pattern Recognition Methods in High Dimensional Settings”, *Pattern Recognition*, 27(8), pp. 1065-1077, 1994