Extension of Minimum Classification Error Training Algorithm

Xuechuan Wang, Kuldip K. Paliwal, Jingdong Chen

***Abstract*—Minimum Classification Error (MCE) training algorithm is a more direct manner to approach the minimum classification error than conventional discriminant methods. However the performance of MCE on a data set with a large number of classes is not satisfactory. The main reason is that the previous definition of misclassification measure can not separate the classes efficiently and MCE is unable to learn the distribution of the classes. In this paper we present some improvements in MCE by giving a new definition to the misclassification measure and using likelihood criteria in discriminant functions.**

1. Introduction

The framework of MCE algorithm was first proposed by Katagiri, Lee and Juang[1]. It is a type of discriminant analysis but achieves the minimum classification error by employing the gradient descent method on a loss function which is a differentiable function of misclassification measure defined as a close approximation of the actual classification error. Thus MCE is a more direct way to achieve the minimum misclassification rate than the conventional discriminative training algorithm. It is so far a powerful tool to design the classifiers and has been used in a number of pattern classification applications[3]. MCE can usually get fairly good classification results on small data sets with few classes and little dimensionality of observation vectors. However, when it is operated on a data set with a large number of classes and a large observation dimensionality, its performance is not satisfactory. The failure of classification of MCE is usually caused by the lack of the knowledge of the actual distribution of the classes. This is because, first, MCE is proposed to be able to optimally separate the task categories even with an incorrect assumption about the data distributions [2,4], little attention has been paid to learn the true distribution of the classes; second, the misclassification measure, which is the key concept of MCE, is only an approximation of classification error and the conventional definition of it can not separate the classes efficiently. Therefore minimizing the misclassification measure does not always lead to good results of classification. This paper presents some improvements in MCE by giving a new definition of the misclassification measure and using the likelihood criteria to make MCE able to learn the distribution of the classes.

2. Extension of MCE training algorithm

Misclassification measure, as the key concept of MCE, is defined based on Bayes decision rule, which is[6],

Decide G = k, if $g(G=k, x) > g(G=j, x)$ for all $j \neq k$.

where $g(G,x)$ is the discriminant function. There are many ways to define the misclassification measure, the commonest way is as follows [2][4][5]:

$$d_k(x^{(p)}) = -g_k(x^{(p)}, \Lambda) + \left[\frac{1}{M-1} \sum_{j \neq k} g_j(x^{(p)}, \Lambda)^{\zeta} \right]^{1/\zeta} \quad (1)$$

where $g_i(x^{(p)}, \Lambda)$, $i = 1, 2, \ldots, M$, is a set of discriminant functions; $x^{(p)}$ is the pth observation vector; M is the number of classes; Λ is the parameter set of each class; ζ is a positive number. One extreme case is when ζ approaches ∞, the misclassification measure becomes:[2]

$$d_k(x^{(p)}) = -g_k(x^{(p)}, \Lambda) + g_i(x^{(p)}, \Lambda) \quad (2)$$

where class i has the largest discriminant value of all the classes other than class k. The loss function is then defined as a monotone function suitable for gradient algorithms to smooth the misclassification measure. Sigmoid function is usually used in the definition of the loss function since it is a smoothed zroe-one function suitable for gradient algorithms, which has the form as follows:

$$L(x^{(p)}) = \frac{1}{1 + e^{-\alpha d(x^{(p)}, \Lambda)}} \quad (3)$$

Since the loss function is only a smooth function, the definition of misclassification measure and the

discriminant function used in it are the two most essential elements in MCE algorithm, which will directly influence the classification performance of the classifier. Our improvements in these two definitions are made as follows.

2.1 Asymptotic Extension of Misclassification measure

The simplest way to define misclassification measure is to define it as the Bayes discriminant for the two-category case:

$$d(x) = P(C_2 \mid x) - P(C_1 \mid x) \tag{4}$$

where $P(C_i|x)$ is the *a posteriori* probabilities. For multi-class cases, the above definition is not suitable in use. Amari[7] proposed a definition in multi-class cases by:

$$d_k(x) = \sum_{i \in S_k} \frac{1}{m_k} [g_i(x,\Lambda) - g_k(x,\Lambda)] \tag{5}$$

where $S_k=\{i \mid g_i(x,\Lambda) > g_i(x,\Lambda)\}$ is the set of confusing classes and m_k is the number of confusing classes in S_k. However, since S_k is not a fixed set, the misclassification measure in (5) is not continuous and deferentiable, thus is not suitable for gradient algorithm. Juang and Katagiri[2] give a definition as (1), and (2) is an extreme case of it. It is obvious that such a definition is suitable for the gradient algorithms. However, since it uses the difference between the correct and its best matching incorrect classes, when gradient algorithm is employed on a certain class, the information of the other class will be treated as the constant and thus lost in differential operation. But when separating a class from its closest class, the information of both classes, especially the centroid information, is very important. Loss of such information may lead to the failure in separating the classes. Meanwhile, separating the classes is essential to MCE algorithm and the failure of it always causes MCE to fail in classification. In order to overcome this defect, we propose a new way to define the misclassification measure. The new definition is as follows:

$$d_k(x^{(p)},\Lambda) = \frac{g_k(x^{(p)},\Lambda)}{\left[\frac{1}{M-1}\sum_{for\ all\ j\neq k} g_j(x^{(p)},\Lambda)^{\zeta}\right]^{1/\zeta}} \tag{6}$$

where the proportion between the discriminant values of correct and incorrect classes is used as the misclassification measure. In the extreme case, i.e. $\zeta \to \infty$, the misclassification measure becomes:

$$d(x^{(p)},\Lambda) = \frac{g_k(x^{(p)},\Lambda)}{g_i(x^{(p)},\Lambda)} \tag{7}$$

Such a definition is able to keep more information of all classes than the definition in (1) and (2) when doing the differential operations and thus suitable for separating the classes. In practice, the performance of MCE based on it is much better than that of MCE based on (1) and (2). The results are given in section 3.

2.2 Extension of the Criterion of the Discriminant Function

In previous MCE algorithm, Euclidean distance and Mahalanobis distance are usually used as the criterion of the discriminant function, whose definitions are:

Euclidean Distance:

$$D_i = \|Y - m^{(i)}\|^2 = \|TX - m^{(i)}\|^2 \tag{8}$$

Mahalanobis Distance:

$$\begin{aligned} D_i &= (Y - m^{(i)})^T \Sigma^{-1} (Y - m^{(i)}) \\ &= (TX - m^{(i)})^T \Sigma^{-1} (TX - m^{(i)}) \end{aligned} \tag{9}$$

where X is the input parameter vector, Y is the transformed feature vector, $m^{(i)}$ is the centroid of class i, T is the transformation matrix and Σ is the covariance matrix of transformed vector Y. The equations for decision boundary between classes are :

$$D_i(x) = D_k(x) \tag{10}$$

However, such a decision boundary is too strict and is difficult to be used to estimate the distribution of classes in the decision space. Here we use the *a posteriori* probability as the discriminant function. Using (7) as the definition of misclassification measure, we get:

$$\begin{aligned} d_k(x^{(p)},\Lambda) &= \frac{P(G=i \mid x^{(p)},\Lambda)}{P(G=k \mid x^{(p)},\Lambda)} \\ &= \frac{P(x^{(p)} \mid G=i,\Lambda)\cdot P(i)}{P(x^{(p)} \mid G=k,\Lambda)\cdot P(k)} \end{aligned} \tag{11}$$

where $P(G \mid x^{(p)}, \Lambda)$ is the *a posteriori* probability; $P(x^{(p)} \mid G, \Lambda)$ is the condition probability; $P(G)$ is the *a priori* probability of class G. Without losing the generality, we suppose that all classes have the same *a priori* probability, then the decision boundary becomes:

$$P(x^{(p)} \mid G=i,\Lambda) = P(x^{(p)} \mid G=k,\Lambda) \tag{12}$$

For the

two-category case, (12) is actually the Bayes decision rule. But it is regarded as an indirect way to define the misclassification measure[2] in multi-category cases because the distribution of the classes is unknown. In fact, by using the *a posteriori* probability as the discriminant function, we can get a more flexible decision boundary and a closer approximation of the distribution of each class.

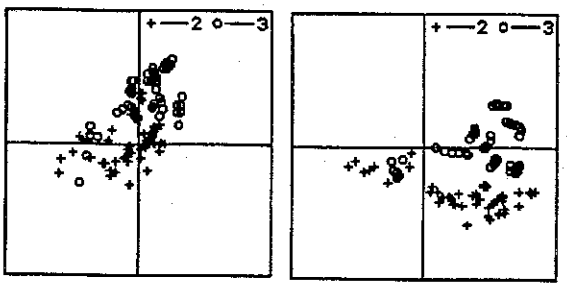
3. Experiments

In this section, we will compare the results of a number of different configurations of MCE algorithms. They are:

1. Using Euclidean distance as the discriminant function and the difference between the correct and incorrect class' discriminant function value as the misclassification measure;
2. Using Mahalanobis distance as the discriminant function and the difference between the correct and incorrect class' discriminant function value as the misclassification measure;
3. Using Euclidean distance as the discriminant function, quotient of the correct and incorrect class' discriminant value as the misclassification measure;
4. Using the *a posteriori* probability as the discriminant function.

The configurations are employed on the Deterding database, which includes 11 vowels and each vowel is represented in vectors of 10 parameters.

Figure 1 shows the distribution of two classes in a two-dimension decision plane. The training vectors have been transformed into such a decision subspace through the transformation matrixes that are obtained by configuration 1 and 3 respectively. The left one is from configuration 1 and the right one is from configuration 3. In the left of Figure 1, it is shown that configuration 1 does not separate the two classes properly. Some part of two classes even overlap each other. Such a distribution of the classes surely will cause the classification to fail. And it does – the correct classification rate is only 39.7727% in the sub-space with dimension 2. The right part of Figure 2 shows that the distribution of the two classes from configuration 3 has a much better separation with each other. Thus the results of classification, which is 67.6136%, are much better in the same sub-space.



Configuration 1 Configuration 2

Figure 1. The distribution of two classes calculated from configuration 1 and 3

Algorithm 2 and 4 are also studied and the results of all the 4 algorithms are compared. Figure 2 shows the results on training data and figure 4 is on testing data. In both figures, the correct classification rate is shown together with the iteration number. It can be seen from Figure 2 that the correct classification rate on training data increases with the number of iteration. But Figure 3 shows the correct classification rate does not. Generally, from these results, we can make the following observations:

1. Definition (6) can separate the classes more efficiently than definition (1). Thus the MCE algorithm based on (6) performs much better than that based on (1).
2. The MCE algorithm with the *a posteriori* probability as the discriminant function performs much better than that with Euclidean or Mahalanobis distance as the discriminant function.

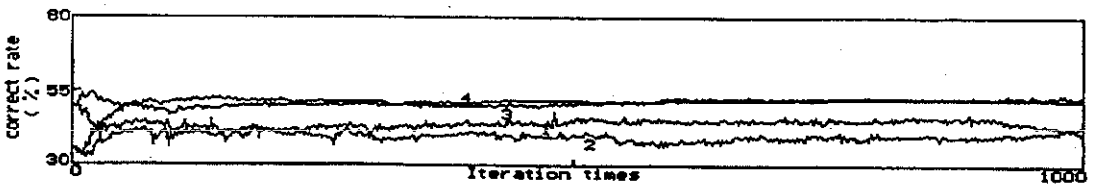


Figure 2. Results on training data

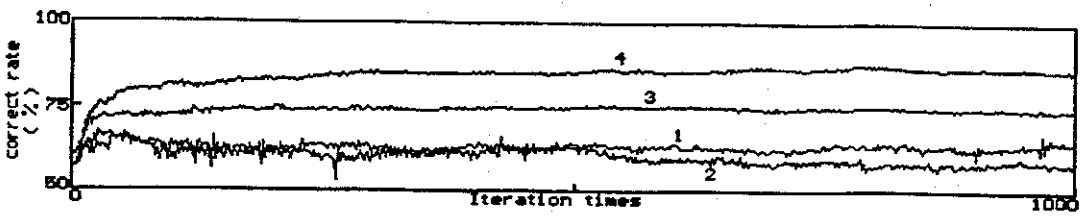


Figure 3. Results on testing data

4. Discussion

Although the improved algorithm achieves much better results than the previous MCE, the recognition score of it on testing data is still very low. From Figure 1, it can be seen that the distribution of some classes is very complicated. In this case, it is hard to train a better model even with a larger training data set because the present MCE algorithm is unable to simulate the distribution precisely. Another difficulty is that MCE training algorithm can not get the actual distribution of the classes since the training data are always insufficient. More work needs to be done in two directions: one is to find a better discriminant function, which can describe the distribution of the classes more precisely, and the other is to use the training data more efficiently so that the algorithm can predict the possible distribution of the classes.

References

[1] Katagiri, S., Lee, C.H., Juang, B.H., "A Generalized Probabilistic Descent Method", Proceeding of the Acoustic Society of Japan, Fall Meeting, 1990, pp 141-142

[2] Biing-Hwang Juang and Shigeru Katagiri, "Discriminative Learning for Minimum Error Classification", *IEEE Trans. On Signal Processing*, vol. 40, pp3043-3054, Dec. 1992

[3] K.K.Paliwal, M.Bacchiani and Y.Sagisaka, "Simultaneous Design of Feature Extractor and Pattern Classification Error Training Algorithm"

[4] Erik McDermott, "New Results for the Prototype-Based Minimum Error Classifier", Preliminary Report, ATR Human Information Processing Research Laboratories, 1994

[5] Erik McDermott and Shigeru Katagiri, "Prototype-Based Minimum Classification Error/Generalized Probabilistic Descent Training for Various Speech Units", *Computer Speech and Language*, Oct. 1994

[6] K.V.Mardia, J.T.Kent and J.M.Bibby, "Multivariate Analysis", Academic Press Inc., San Diego, 1979

[7] S. Amari, "A Theory of Adaptive Pattern classifiers", IEEE Trans. On Electronic Compution, vol. 16, pp299-307, Jun. 1997