# MINIMUM CLASSIFICATION ERROR TRAINING ALGORITHM FOR FEATURE EXTRACTOR AND PATTERN CLASSIFIER IN SPEECH RECOGNITION

*K.K. Paliwal, M. Bacchiani and Y. Sagisaka*

ATR Interpreting Telecommunications Res. Labs.
2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

## ABSTRACT

Recently, a minimum classification error training algorithm has been proposed for minimizing the misclassification probability based on a given set of training samples using a generalized probabilistic descent method. This algorithm is a type of discriminative learning algorithm, but it approaches the objective of minimum classification error in a more direct manner than the conventional discriminative training algorithms. We apply this algorithm for simultaneous design of feature extractor and pattern classifier, and demonstrate some of its properties and advantages.

## 1. INTRODUCTION

Juang and Katagiri [1] have recently proposed a minimum classification error training algorithm which minimizes the misclassification probability based on a given set of training samples using a generalized probabilistic descent method. This algorithm is a type of discriminative learning algorithm, but it approaches the objective of minimum classification error in a more direct manner than the conventional discriminative training algorithms [2]. Because of this, it has been used in a number of pattern classification applications [3, 4, 5, 6, 7, 8]. For example, Chang et al. [3] and Komori and Katagiri [4] have used this algorithm for designing the pattern classifier for dynamic time-warping based speech recognition, Chou et al. [5] and Rainton and Sagayama [6] for hidden Markov model (HMM) based speech recognition, Sukkar and Wilpon [7] for word spotting, and Liu et al. [8] for HMM-based speaker recognition. More recently, the minimum classification error training algorithm has been used for feature extraction [9, 10, 11, 12]. For example, Biem and Katagiri have used it for determining the parameters of a cepstral lifter [9] and a filter bank [10]. They have found that the resulting parameters of cepstral lifter and filter bank have a good physical interpretation. Bacchiani and Aikawa [11] have used this algorithm for designing a dynamic cepstral filter. Watanabe and Katagiri [12] have used a class-dependent unitary transformation for feature extraction whose parameters are determined by the minimum classification error training algorithm.

In the present paper, we use the minimum classification error (MCE) training algorithm to design the feature extractor as well as the classifier. Consider a canonic pattern recognizer shown in Fig. 1. The aim of the pattern recognizer is to classify an input signal as one of the K classes. This is done in two steps: a feature analysis step and a pattern classification step. In the feature analysis step, the input signal is analyzed and $D$ parameters are measured. The D-dimensional parameter vector $X$ is processed by the feature extractor which produces at its output a feature vector $Y$ of dimensionality $\leq D$. In our study, we use a $D \times D$ linear transformation $T$ for feature extraction; i.e., $Y = TX$. The transformation $T$ is a general linear transformation; i.e., it is not restricted, for example, to be unitary as done by Watanabe and Katagiri [12]. The feature vector $Y$ is a D-dimensional vector in our study. In the pattern classification step, the feature vector is compared with each of the K class-models and a dissimilarity (or, distance) measure is computed for each class. The class that gives the minimum distance is considered to be the recognized class.

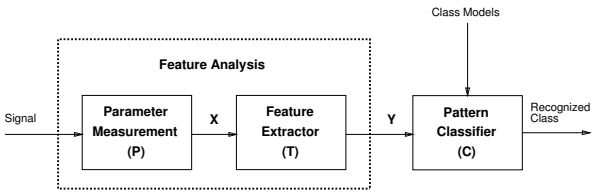


Figure 1: A pattern recognition system.

In the present paper, we study the minimum classification error training algorithm for the following configurations of feature extractor and classifier:

- **Configuration 1:** It is a baseline configuration where transformation $T$ is unity and the class models are determined by the maximum-likelihood (ML) training algorithm (as the class-dependent means of the training vectors).
- **Configuration 2:** Here the transformation $T$ is kept fixed to a unity matrix and the class models are computed using the MCE training algorithm.
- **Configuration 3:** Here the transformation $T$ is computed by the MCE training algorithm and the class models are kept fixed to the baseline class models.
- **Configuration 4:** This configuration is similar to Configuration 3, except that transformation $T$ is applied to the parameter vector as well as to the class models of the baseline configuration prior to distance computation.
- **Configuration 5:** Here both the transformation $T$ and the class models are computed independently using the MCE training algorithm.

K.K. Paliwal's present address: School of Microelectronic Engineering, Griffith University, Brisbane, QLD 4111, Australia

- **Configuration 6:** This configuration is similar to Configuration 3, except that the transformation T is made class-dependent; i.e., we now have K different transformations, $T_k$, $k = 1, 2, \ldots, K$ for K different classes. We apply transformation $T_k$ to parameter vector $X$ before computing its distance from the k-th class.
- **Configuration 7:** This configuration is similar to Configuration 5, except that the transformation T is made class-dependent (similar to Configuration 6); i.e., we now have K different transformations, $T_k$, $k = 1, 2, \ldots, K$ for K different classes.

Note that Configurations 2,3,4 and 5 use a class-independent transformation as shown in Fig. 2; while Configurations 6 and 7 use a class-dependent transformation as shown in Fig. 3.

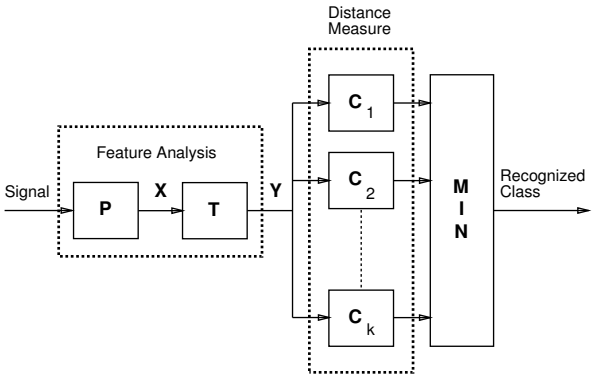


Figure 2: A pattern recognition system with class-independent transformation.

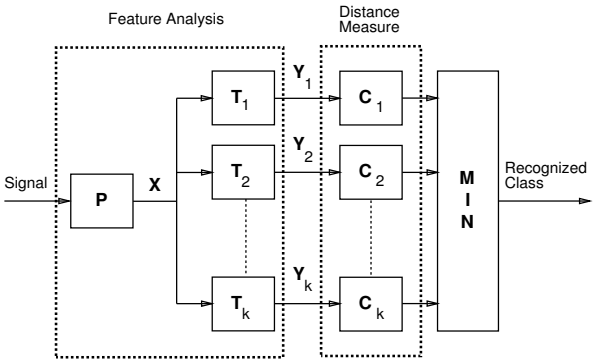


Figure 3: A pattern recognition system with class-dependent transformation.

## 2. MCE TRAINING ALGORITHM

In this section, we describe briefly the minimum classification error (MCE) training algorithm. For more details, see [1]. The MCE algorithm is described here only for Configuration 5. It can be extended to other configurations in a straightforward manner.

In the pattern recognition system shown in Fig. 1, the input parameter vector $X$ is transformed to a feature vector $Y$ $(= TX)$ and classified into class $i$ if

$$D_i \leq D_j, \qquad \text{for all } j \neq i, \tag{1}$$

where $D_i$ is the distance of feature vector $Y$ from class $i$. In the present paper, we use a simple Euclidean distance measure to define this distance. It is given by

$$\begin{aligned} D_i &= \| Y - m^{(i)} \|^2 \\ &= \| TX - m^{(i)} \|^2, \end{aligned} \tag{2}$$

where $m^{(i)}$ is the prototype vector representing the class $i$.

Here, we are given a total of $P$ labeled training vectors; i.e., we have $P$ parameter vectors $X^{(1)}, X^{(2)}, \ldots, X^{(P)}$ available for training with corresponding classifications $C^{(1)}, C^{(2)}, \ldots, C^{(P)}$ known to us. Our aim here is to use the MCE algorithm to estimate the transformation matrix $T$ and class prototype vectors $m^{(1)}, m^{(2)}, \ldots, m^{(K)}$ using these labeled training vectors. The procedure for doing this is described below.

The distance of $p$th training vector from class $i$ is given by

$$\begin{aligned} D_i^{(p)} &= \| Y^{(p)} - m^{(i)} \|^2 \\ &= \| TX^{(p)} - m^{(i)} \|^2 \\ &= \sum_{s=1}^{D} \left( \sum_{j=1}^{D} T_{sj} X_j^{(p)} - m_s^{(i)} \right)^2. \end{aligned} \tag{3}$$

We use this distance to define the misclassification measure for the $p$th training vector as follows:

$$d^{(p)} = D_{C^{(p)}}^{(p)} - D_{N^{(p)}}^{(p)}, \tag{4}$$

where $D_{C^{(p)}}^{(p)}$ is the distance of $p$th training vector from its known class $C^{(p)}$ and the distance $D_{N^{(p)}}^{(p)}$ is computed from the relation

$$D_{N^{(p)}}^{(p)} = \underset{i, i \neq C^{(p)}}{\operatorname{argmin}} D_i^{(p)}. \tag{5}$$

The loss function $L^{(p)}$ for the $p$th training vector is then defined as the sigmoid of the misclassification measure as follows:

$$\begin{aligned} L^{(p)} &= f(d^{(p)}) \\ &= \frac{1}{1 + e^{-\alpha d^{(p)}}}, \end{aligned} \tag{6}$$

where $\alpha$ is a parameter defining the slope of the sigmoid function.

The total loss function $L$ is defined as

$$L = \sum_{p=1}^{P} L^{(p)}. \tag{7}$$

In the MCE algorithm, the transformation matrix and class prototype vectors are obtained by minimizing this loss function through the steepest gradient descent algorithm. This is an iterative algorithm where parameters at the $(k+1)$th iteration are computed from the $k$th iteration results as follows:

$$T_{sj}(k+1) = T_{sj}(k) - \eta \frac{\partial L}{\partial T_{sj}}, \tag{8}$$

$$m_s^{(N^{(p)})}(k+1) = m_s^{N^{(p)}}(k) - \eta \frac{\partial L}{\partial m_s^{(N^{(p)})}}, \tag{9}$$

and

$$m_s^{(C^{(p)})}(k+1) = m_s^{C^{(p)}}(k) - \eta \frac{\partial L}{\partial m_s^{(C^{(p)})}}, \qquad (10)$$

where $\eta$ is a positive constant (known as the adaptation constant) and

$$\frac{\partial L}{\partial T_{sj}} = 2\alpha \sum_{p=1}^{P} f(d^{(p)})(1 - f(d^{(p)}))(m_s^{(N^{(p)})} - m_s^{(C^{(p)})})X_j^{(p)}, \qquad (11)$$

$$\frac{\partial L}{\partial m_s^{(N^{(p)})}} = 2\alpha \sum_{p=1}^{P} f(d^{(p)})(1 - f(d^{(p)}))(\sum_{j=1}^{D} T_{sj}X_j^{(p)} - m_s^{(N^{(p)})}), \qquad (12)$$

and

$$\frac{\partial L}{\partial m_s^{(C^{(p)})}} = -2\alpha \sum_{p=1}^{P} f(d^{(p)})(1 - f(d^{(p)}))(\sum_{j=1}^{D} T_{sj}X_j^{(p)} - m_s^{(C^{(p)})}). \qquad (13)$$

For the initialization of the MCE algorithm, 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 class-conditioned means).

## 3. RESULTS

The MCE algorithm is studied here on a multispeaker vowel recognition task. The Peterson-Barney vowel data base [13] is used for this purpose. Here each vowel is represented in terms of 4 parameters: fundamental frequency and frequencies of the first three formants. The data base consists of two repetitions of 10 vowels in /hVd/ context recorded from 76 speakers (33 men, 28 women and 15 children). Fundamental and formant frequencies were measured by Peterson and Barney from the central steady-state portions of the /hVd/ utterances. We use the first repetition for training and the second for testing. We use the Euclidean distance measure for classification of the 4-dimensional parameter vector into one of the 10 classes. The model for each class is determined as the mean vector of the training patterns of that class. In our implementation of the MCE training algorithm, we use the steepest gradient descent algorithm with the adaptation parameter updated every iteration using a fast-converging algorithm described by Lucke [14].

In order to study the convergence properties of this algorithm, we study it for Configuration 5. Figure 4 shows the the loss function, recognition error on training and test data as a function of iteration number. It can be seen from this figure that the loss function is decreasing with number of iterations and the algorithm is converging reasonably fast (within 500 iterations). Also, recognition results on test data are similar to those on training, showing the generalization property of the algorithm.

The MCE algorithm is studied for all the seven configurations. The results for different configurations are listed in Table 1. We list in column 2 of this table the total number of free parameters used in the transformation and the classifier. Column 3 of this table lists the number

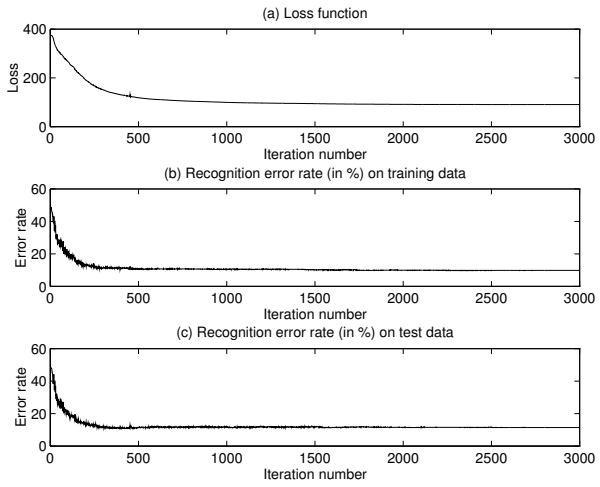


Figure 4: Results for Configuration 5 as a function of iteration number. (a) Loss function, (b) Recognition error rate (in %) on training data, and (c) Recognition error rate (in %) on test data.

of parameters computed by the MCE training algorithm. The numbers shown within square brackets in columns 2 and 3 correspond to the vowel recognition task used in this study, where $K = 10$ and $D = 4$. From this table, we can make the following observations:

1. The MCE training algorithm performs better than the ML algorithm (compare Configuration 1 with Configuration 2).

2. The recognition performance of the pattern recognizer improves with an increase in the total number of free parameters in the transformation and the classifier.

3. For a given number of free parameters, the recognition performance improves as the more number of parameters are updated by the MCE training algorithm (compare Configuration 3 with Configuration 5 or Configuration 6 with Configuration 7).

4. Observe Configurations 3 and 4. Both of these configurations have same number of free parameters and same number of parameters are updated by the MCE training algorithm. But, Configuration 4 gives significantly better results than Configuration 3. This is because in Configuration 4 the transformation $T$ is applied to the parameter vector $X$ as well as the class models prior to distance computation.

5. Observe Configurations 5 and 7. Configuration 5 uses a class independent transformation, while Configuration 7 uses class-dependent transformations. Therefore, Configuration 7 shows better recognition performance than Configuration 5 on training data, though the difference in the recognition rates for the two configurations is small. However, note that recognition performance of Configuration 7 on test data is inferior to that of Configuration 5. This happens because the number of parameters updated by the MCE algorithm are too large for the limited amount of data available for training and, hence, the results do not generalize to test data properly.

Table 1: Recognition error rate (in %) for different configurations of feature extractors and classifiers studied using the minimum classification error (MCE) training algorithm.

| Configuration | Total no. of free parameters | No. of parameters updated by MCE | Recognition error rate | |
|---|---|---|---|---|
| | | | Training | Test |
| Conf. 1 | $KD$ [40] | 0 [0] | 48.29 | 48.16 |
| Conf. 2 | $KD$ [40] | $KD$ [40] | 34.47 | 36.18 |
| Conf. 3 | $D^2 + KD$ [56] | $D^2$ [16] | 33.16 | 33.29 |
| Conf. 4 | $D^2 + KD$ [56] | $D^2$ [16] | 13.95 | 16.32 |
| Conf. 5 | $D^2 + KD$ [56] | $D^2 + KD$ [56] | 9.87 | 11.45 |
| Conf. 6 | $K(D^2 + D)$ [200] | $KD^2$ [160] | 9.47 | 12.37 |
| Conf. 7 | $K(D^2 + D)$ [200] | $K(D^2 + D)$ [200] | 9.34 | 12.76 |

## 4. SUMMARY

In this paper, we have studied the use of minimum classification error (MCE) training algorithm for the design of the feature extractor and the pattern classifier. We have investigated a number of configurations of the feature extractor and the pattern classifier, and have demonstrated a number of properties and advantages of the MCE training algorithm.

## REFERENCES


[1] B.H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, Vol. 40, No. 12, pp. 3043-3054, Dec. 1992.

[2] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[3] P.C. Chang, S.H. Chen and B.H. Juang, "Discriminative analysis of distortion sequences in speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1991, Vol. 1, pp. 549-552.

[4] T. Komori and S. Katagiri, "GPD training of dynamic programming-based speech recognizers," *Journal of Acoust. Soc. of Japan (E)*, Vol. 13, No. 6, pp. 341-349, Nov. 1992.

[5] W. Chou, B.H. Juang and C.H. Lee, "Segmental GPD training of HMM based speech recognizer," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Mar. 1992, Vol. 1, pp. 473-476.

[6] D. Rainton and S. Sagayama, "Minimum error classification training of HMMs – Implementation details and experimental results," *Journal of Acoust. Soc. of Japan (E)*, Vol. 13, No. 6, pp. 379-387, Nov. 1992.

[7] R.A. Sukkar and J.G. Wilpon, "A two-pass classifier for utterance rejection in keyword spotting," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1993, Vol. 2, pp. 451-454.

[8] C.S. Liu, C.H. Lee, W. Chou, B.H. Juang and A.E. Rosenberg, "A study on minimum error discriminative training for speaker recognition," *Journal of Acoust. Soc. of Am.*, Vol. 97, No. 1, pp. 637-648, Jan. 1995.

[9] A. Biem and S. Katagiri, "Feature extraction based on minimum classification error/generalized probabilistic descent method," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1993, Vol. 2, pp. 275-278.

[10] A. Biem and S. Katagiri, "Filter bank design based on discriminative feature extraction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1994, Vol. 1, pp. 485-488.

[11] M. Bacchiani and K. Aikawa, "Optimization of time-frequency masking filters using the minimum classification error criterion," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1994, Vol. 2, pp. 197-200.

[12] H. Watanabe, T. Yamaguchi and S. Katagiri, "Discriminative metric design for pattern recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1995, Vol. 5, pp. 3439-3442.

[13] G. Peterson and H.L. Barney, "Control methods used in a study of the vowels," *Journal of Acoust. Soc. of Am.*, Vol. 24, pp. 175-184, 1952.

[14] H. Lucke, "On the representation of temporal data for connectionist word recognition," Ph.D. Thesis, Cambridge University, Nov. 1991.