GPD-Based State Modification by Weighted Linear Loss Function

Taehee Kwon and Hanseok Ko

Department of Electronics Engineering, Korea University
5 ka-1, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
thkwon@ispl.korea.ac.kr, hsko@korea.ac.kr

Abstract. This paper proposes an effective loss function to assign the HMM state weight based on the MCE method. This is to remedy the performance limitation inherent in the traditional maximum likelihood method, which adjusts parameters to maximize the likelihood of training HMM. If minimum classification error method is used to minimize the error rate for training data, then the local optimum point can be achieved and the recognition performance can be achieved. However, if the amount of data used in the MCE training is too small, there can be a risk of overfitting to the training data. In this paper, we propose state modification by weighted linear loss function to overcome overfitting to training data. Representative experiments confirm this postulation and show the improvement in error rate when applied.

1 Introduction

Hidden Markov modeling (HMM) has become prevalent in speech recognition for capturing the acoustic models. It is statistically based and links a recognition task to the distribution estimation. The most commonly used method out of these distribution estimation methods is the maximum likelihood (ML) estimation. However, we lack the complete knowledge on the form of data distribution and training data is always inadequate in dealing with speech recognition. Usually the performance of a recognizer is normally defined by its expected recognition error rate and an optimal recognizer is the one that achieves the least expected recognition error rate. In this perspective, we can propose the state modification to exploit the discriminant information latent in the state likelihood without improved feature extraction, improved acoustic resolution. However, if the ML estimated HMMs have the high performance to some extent and the number of recognition errors used in the minimum classification error (MCE) training is small, the overfitting problem occurs as we train iteratively using the training data. Therefore, the MCE training not for the full training set but for misrecognized training set results in overfitting to small misrecognized training set. In this circumstance, there's a need to increase the number of errors by adjusting the misclassification measure. If we define a new loss function and train HMMs based on this loss function, then we can expect that overtraining can be prevented and the recognition error rate is reduced by producing increased number of recognition errors and adapting the state weights.

This paper begins in section 2 with a brief review of the MCE method. We then discuss in section 3 the state modification based on the MCE method. We discuss the newly proposed loss function in section 4 and discuss experimental results about state modification using four different types of loss functions in section 5. Finally, conclusion is described in section 6.

2 MCE Training

The aim of MCE training is to correctly discriminate the observations of an HMM for the best recognition results and not necessarily to try and find the best model for the distributions of the data. This section will briefly describe and discuss the MCE algorithm [1][2][3][4][5]. We shall also define an optimization criterion, which provides a reasonable estimate of the error probability.

In the traditional HMM-based speech recognizer, the discriminant function for class i is defined by the following equation for pattern classification

$$g_i(\mathbf{X}; \Lambda) = \sum_{t=1}^T \left[\log a_{\bar{q}_{t-1}\bar{q}_t}^{(i)} + \log b_{\bar{q}_t}^{(i)}(\mathbf{x}_t) \right] + \log \pi_{\bar{q}_0}^{(i)} \quad (1)$$

where Λ is a set of classifier parameters, $\mathbf{X}$ is an observation sequence, $\bar{\mathbf{q}} = (\bar{q}_0, \bar{q}_1, \dots, \bar{q}_T)$ is the optimal state sequence that maximizes the joint state-observation function for class i , a_{ij} denotes the probability of transition from state i to state j , and $b_j(\mathbf{x}_t)$ denotes the probability density function of observing $\mathbf{x}_t$ at state j . In a continuous multivariate mixture Gaussian HMM, the state output distribution is as follows.

$$b_j(\mathbf{x}_t) = \sum_{m=1}^M c_{jm} N(\mathbf{x}_t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) \quad (2)$$

where $N(\cdot)$ denotes a multivariate Gaussian density, $\boldsymbol{\mu}_{jm}$ is the mean vector in state j , mixture m and $\boldsymbol{\Sigma}_{jm}$ is the covariance matrix in state j , mixture m .

For input utterance, the decision rule is used. For an input utterance $\mathbf{X}$, the class C_i is decided as

$$C(\mathbf{X}) = C_i \quad \text{if } i = \arg \max_j g_j(\mathbf{X}; \Lambda) \quad (3)$$

where $g_j(\mathbf{X}; \Lambda)$ is the discriminant function of the input utterance or observation sequence $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ for the j th model.

In first, it is necessary to express the operational decision rule (3) in a functional form. A class misclassification measure which is a continuous function of the classifier parameters Λ and attempts to emulate the decision rule is therefore defined.

$$d_i(\mathbf{X}; \Lambda) = -g_i(\mathbf{X}; \Lambda) + \log \left[\frac{1}{N} \sum_{j=1, j \neq i}^N \exp[g_j(\mathbf{X}; \Lambda) \eta] \right]^{\frac{1}{\eta}} \quad (4)$$

where η is a positive constant and N is the number of N -best competing classes. For an i th class utterance $\mathbf{X}$, $d_i(\mathbf{X}) > 0$ implies misclassification and $d_i(\mathbf{X}) \leq 0$ means correct classification.

The complete loss function is defined in terms of the misclassification measure using a smooth zero-one function.

$$l_i(\mathbf{X}; \Lambda) = l(d(\mathbf{X}; \Lambda)) \quad (5)$$

The smooth zero-one function can be any continuous zero-one function, but is typically the following sigmoid function

$$l(d) = \frac{1}{1 + \exp[-rd + \theta]} \quad (6)$$

where θ is usually set zero or slightly smaller than zero and r is a constant. Finally, for any unknown $\mathbf{X}$, the classifier performance is measured by

$$l(\mathbf{X}; \Lambda) = \sum_{i=1}^M l_i(\mathbf{X}; \Lambda) \mathbf{1}(\mathbf{X} \in C_i) \quad (7)$$

where $\mathbf{1}(\cdot)$ is the indicator function.

The optimal classifier parameters are those that minimize the expected loss function. The generalized probabilistic descent (GPD) algorithm is used to minimize the expected loss function. The GPD algorithm is given by

$$\Lambda_{n+1} = \Lambda_n - \varepsilon_n \mathbf{U}_n \nabla l(\mathbf{X}; \Lambda) |_{\Lambda=\Lambda_n} \quad (8)$$

where $\mathbf{U}$ is a positive definite matrix, ε_n is the learning rate or step size of adaptation, and Λ_n is the classifier parameter set at time n .

The GPD algorithm is an unconstrained optimization technique. But some constraints must be maintained for HMMs so some modifications are required. Instead of using a complicated constrained GPD algorithm, Chou et al [1], applied GPD to transformed HMM parameters. The parameter transformations ensure that there are no constraints in the transformed space where the updates occur. The following HMM constraints should be maintained in the original space.

$$\sum_j a_{ij} = 1 \text{ and } a_{ij} \geq 0, \quad \sum_k c_{jk} = 1 \text{ and } c_{jk} \geq 0, \quad \sigma_{jkl} \geq 0 \quad (9)$$

The following parameter transformations should be used before and after parameter adaptation.

$$\begin{aligned}
a_{ij} &\rightarrow \tilde{a}_{ij} \quad \text{where } a_{ij} = e^{\tilde{a}_{ij}} / \left(\sum_k e^{\tilde{a}_{ik}} \right) \\
c_{jk} &\rightarrow \tilde{c}_{jk} \quad \text{where } c_{jk} = e^{\tilde{c}_{jk}} / \left(\sum_k e^{\tilde{c}_{jk}} \right) \\
\mu_{jkl} &\rightarrow \tilde{\mu}_{jkl} = \mu_{jkl} / \sigma_{jkl} \\
\sigma_{jkl} &\rightarrow \tilde{\sigma}_{jkl} = \log \sigma_{jkl}
\end{aligned} \tag{10}$$

3 State Modification

In order to exploit the discriminative information latent in state likelihood, each state have a weight such that each state output score is a product of state output likelihood and state weight. We assume that there are M HMMs as the basic recognition units, and that the total number of states is J in a recognition unit. The basic recognition units consist of phonemes or words.

In the traditional HMMs, the discriminant function for class i is defined by equation (1). In contrast, in the state modified HMMs, the discriminant function for class i is denoted by the following equation (11)

$$g_i(\mathbf{X}; \Lambda) = \sum_{t=1}^T \left[\log a_{\bar{q}_{t-1}\bar{q}_t}^{(i)} + w_{\bar{q}_t}^{(i)} \log b_{\bar{q}_t}^{(i)}(\mathbf{x}_t) \right] + \log \pi_{\bar{q}_0}^{(i)} \tag{11}$$

where $w_{\bar{q}_t}^{(i)}$ is the weight of a optimal state at time t in class i .

Adaptation of state weights set initially to one is given by previously discussed equations (4) ~ (7) and equation (12).

$$w_j^{(i)}(n+1) = w_j^{(i)}(n) - \varepsilon_n \frac{\partial}{\partial w_j^{(i)}} l(\mathbf{X}; \Lambda) \Big|_{w_j^{(i)} = w_j^{(i)}(n)} \tag{12}$$

Similar to equation (9), the state weight constraints should be maintained in the original space. In other words, the state weights for a speech recognition unit are constrained by equation (13)

$$\sum_{j=1}^J w_j = J, \quad 0 < w_j < J \tag{13}$$

where w_j is a state weight for state j in a HMM and J is the number of states in a HMM.

Finally, parameter transformations are given by the following equations.

$$w_j \rightarrow \tilde{w}_j \quad \text{where } w_j = e^{\tilde{w}_j} / \left(\sum_k e^{\tilde{w}_k} \right) \tag{14}$$

where $\tilde{w}_j$ is the transformed state weight of w_j .

In the following sections, we propose four different kinds of loss functions and discuss experimental results of applying those loss functions for state modification.

4 Linear Loss Function

In general, we only consider misclassified speech in the MCE training. However, if the amount of training data used in MCE training is too small and we just focus on this misrecognized training data, then the overfitting to training data occurs and there exists the difference between training and testing set error rates. So there is a need to modify the smooth zero-one loss function that we should minimize into an alternative form. For this purpose, we propose alternative loss functions. For the first time, we can modify the misclassification measure by adding a weighted likelihood of correct class to the misclassification measure [6]. This modified misclassification measure is inserted into a sigmoid function to produce the sigmoid zero-one loss function. In the next time, we can consider a misclassification measure as a loss function to produce the linear loss function. By using this linear loss function, we can increase the gradient associated with a loss function for correct string by a uniform factor k while not affecting the gradient associated with a loss function for incorrect string as shown in Equation (16).

$$d_i(\mathbf{X}; \Lambda) = -g_i(\mathbf{X}; \Lambda) + \log \left[\frac{1}{N} \sum_{j=1, j \neq i}^N \exp[g_j(\mathbf{X}; \Lambda)\eta] \right]^{\frac{1}{\eta}} \quad (15)$$

$$\begin{aligned} \tilde{d}_i(\mathbf{X}; \Lambda) &= d_i(\mathbf{X}; \Lambda) - k \cdot g_i(\mathbf{X}; \Lambda) \\ &= -(1+k) \cdot g_i(\mathbf{X}; \Lambda) + \log \left[\frac{1}{N} \sum_{j=1, j \neq i}^N \exp[g_j(\mathbf{X}; \Lambda)\eta] \right]^{\frac{1}{\eta}} \end{aligned} \quad (16)$$

As a result of modified misclassification measure, another loss functions are sigmoid zero-one loss function where a modified misclassification measure is inserted into a sigmoid function, weighted linear loss function that is exactly the same as a misclassification measure (16).

The weighted linear loss function will tend to reinforce correct strings efficiently while still penalizing errors such that this weighted linear loss function can reduce overfitting to training data by adjusting the value of k .

5 Experiments

To investigate the performance of the proposed method in speech recognition, we used the Korean isolated digit database recorded in a silent room. The isolated digit database consists of utterances uttered by 500 speakers (250 male and 250 females). Utterances from 400 speakers (200 males and 200 females) were used as training data

and those from 100 speakers (50 males and 50 females) were used as testing data. Each speech sampled by 16 KHz sampling rate was down sampled into 11.025 KHz rate for implementing a speech recognizer on a DSP board. We extract 13-order mel-frequency cepstral coefficients (MFCCs), 13-order delta coefficients, 13-order acceleration coefficients including log energy. As a result, an observation consists of a 3 stream, 39-order feature vector. We chose the phoneme unit as a recognition unit and produced a 3-state, 8 mixture HMM for each phoneme unit. All state output distributions consist of 8 mixture, multivariate mixture Gaussian distributions. We constructed a HMM set as baseline system by ML method and then constructed four HMM sets by weighting state output likelihood based on GPD method using different loss functions. First of all, we perform the MCE training of basic parameters to compare performance by state modification to performance by MCE training of basic parameters.

5.1 MCE Training of Basic Parameters

In this section, we performed the MCE training of basic HMM parameters using four loss functions. The basic HMM parameters mean mixture weights, means, standard deviations of each state output distribution. Table 1 shows that training and testing data accuracies vary when the accuracy of training data reached the maximum point. As the number of iterations increases, the accuracy of training data is increased gradually but the accuracy of testing data is not increased efficiently. The aim of MCE training is to correctly discriminate the observations of an HMM for the best recognition results of training and testing data. So, the MCE training of basic parameters is not efficient in this case.

Table 1. Word accuracies (%) with four kinds of loss functions

Data	MLE	SIG	WL+SIG	LIN	WL+LIN
Training	98.40	98.58	98.58	98.58	98.71
Testing	98.00	98.00	98.00	98.00	98.00

SIG : sigmoid loss function, LIN : linear loss function

WL : weighted likelihood of correct class

5.2 MCE Training of State Weights

In this section, we performed the MCE training of HMM state weights using four loss functions. Each state weights are set to one as initial values. To estimate the state weights, we used 3 most confusable strings for computing the misclassification measure, $N = 3$ and set $r = 0.01$, $k = 0.005$.

We performed four experiments based on four different loss functions i.e. sigmoid zero-one loss function, sigmoid zero-one loss function resulting from a modified misclassification measure, a linear loss function, a weighted linear loss function resulting from a modified misclassification measure. Table 2 shows maximum word accuracies of both training and testing data when the word accuracy of testing data reaches the maximum point in the four different experiments.

Table 2. Word accuracies (%) with four kinds of loss functions

Data	MLE	SIG	WL+SIG	LIN	WL+LIN
Training	98.40	98.54	98.54	98.44	98.56
Testing	98.00	98.17	98.58	98.00	98.50

Fig. 1 shows that training and testing data accuracies vary as the number of iterations increases using a sigmoid zero-one loss function resulting from a misclassification measure. As the number of iterations increases, both training and testing data accuracies decrease owing to overfitting to training data. The aim of MCE training is to correctly discriminate the observations of an HMM for the best recognition results. Therefore, this result is not very desirable. However, Fig. 2 shows that the training data accuracy decreases gradually while the testing data accuracy fluctuates significantly as the number of iterations increase.

Fig. 1. MCE Training of State Weights (SIG)

Fig. 2. MCE Training of State Weights (WL+SIG)

Fig. 3 shows that training and testing data accuracies are not invariant nearly as the number of iterations increases using a linear loss. However, Fig. 4 shows that training and testing data accuracies increase consistently at the same time using a weighted linear loss function resulting from the modified misclassification measure as the number of iterations increases. So, this weighted linear loss function with weighted likelihood term is very efficient to overcome overtraining in state modification and the word accuracy is comparable to the ML method without improved feature extraction, improved acoustic resolution.

Fig. 3. MCE Training of State Weights (LIN)

Fig. 4. MCE Training of State Weights (WL+LIN)

It is important to adjust k -values suitably in state modification by weighted linear loss function. Therefore, the performance improvements as the result of adjusting the values of k are shown in Table 3. Table 3 shows word accuracies of training and testing data in the case where the accuracy of testing data reaches the local maximum point.

Table 3. Word accuracies (%) as the result of adjusting the values of k

k	0.001	0.002	0.003	0.004	0.005
Training	98.54	98.63	98.48	98.67	98.56
Testing	98.17	98.25	98.25	98.25	98.50

6 Conclusions

Experimental results showed that the use of weighted linear loss function by weighted likelihood of correct class could improve word accuracies for both training and testing data consistently. It is very important to reduce word accuracies for both training and testing data consistently. Fig. 4 shows the consistency of performance improvement in training and testing data and Table 2 shows that the state modification by weighted linear loss function with weighted likelihood of correct class reduces word error rate by 25 % comparing to ML method. In this result, the state modification by weighted linear loss function is very efficient to discriminate observations.

References

1. X.Huang, et al: Spoken Language Processing, Prentice-Hall PTR, 2001.
2. Biing-Hwang Juang, Wu Chou, Chin-Hui Lee: Minimum Classification Error Rate Methods for Speech Recognition, IEEE Transactions on Speech and Audio Processing, Vol. 5, No. 3, PP. 257–265, May 1997.
3. Wu Chou, Chin-Hui Lee, Biing-Hwang Juang: Minimum Error Rate Training Based on N-best String Models, IEEE ICASSP 93, PP. 652–655, April 1993.
4. Wu Chou: Discriminant-Function-Based Minimum Recognition Error Rate Pattern-Recognition Approach to Speech Recognition, Proceedings of the IEEE, Vol. 88, No. 8, PP. 1201–1223, August 2000.
5. Min-Tau Lin, Andreas Spanias, Philipos Loizou: An Improved Approach to Robust Speech Recognition Using Minimum Error Classification, Speech Communication, Vol.30, PP. 27–36, 2000.
6. Darryl William Purnell and Elizabeth C. Botha: Improved Generalization of MCE Parameter Estimation with Application to Speech Recognition, IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 4, PP. 232–239, May 2002.
7. O.W. Kwon, C.K. Un: Performance of HMM-based Speech Recognizers with Discriminative State-Weights, Speech Communication, Vol. 19, PP. 197–205, June 1996.
8. S. J. Young: The HTK Book 3.0, Cambridge University, 2000.