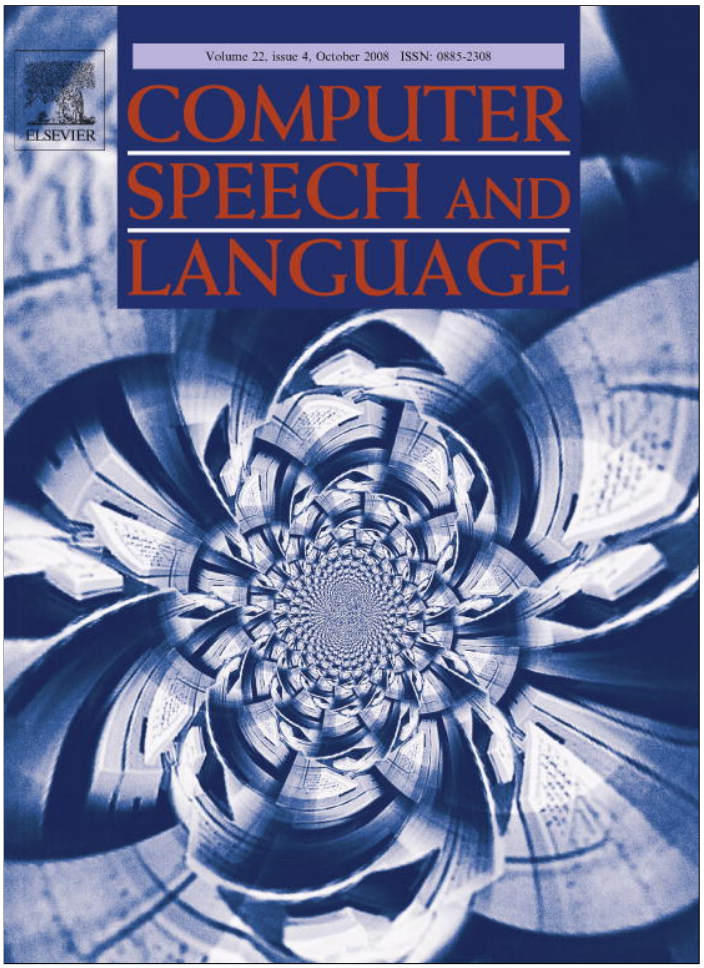

Available online at www.sciencedirect.com
ScienceDirect
Computer Speech and Language 22 (2008) 415–429
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

Large-margin minimum classification error training: A theoretical risk minimization perspective ☆

Dong Yu *, Li Deng, Xiaodong He, Alex Acero

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA

Received 17 May 2007; received in revised form 4 March 2008; accepted 5 March 2008
Available online 12 March 2008

Abstract

Large-margin discriminative training of hidden Markov models has received significant attention recently. A natural and interesting question is whether the existing discriminative training algorithms can be extended directly to embed the concept of margin. In this paper, we give this question an affirmative answer by showing that the sigmoid bias in the conventional minimum classification error (MCE) training can be interpreted as a soft margin. We justify this claim from a theoretical classification risk minimization perspective where the loss function associated with a non-zero sigmoid bias is shown to include not only empirical error rates but also a margin-bound risk. Based on this perspective, we propose a practical optimization strategy that adjusts the margin (sigmoid bias) incrementally in the MCE training process so that a desirable balance between the empirical error rates on the training set and the margin can be achieved. We call this modified MCE training process large-margin minimum classification error (LM-MCE) training to differentiate it from the conventional MCE. Speech recognition experiments have been carried out on two tasks. First, in the TIDIGITS recognition task, LM-MCE outperforms the state-of-the-art MCE method with 17% relative digit-error reduction and 19% relative string-error reduction. Second, on the Microsoft internal large vocabulary telephony speech recognition task (with 2000 h of training data and 120 K words in the vocabulary), significant recognition accuracy improvement is achieved, demonstrating that our formulation of LM-MCE can be successfully scaled up and applied to large-scale speech recognition tasks.

Keywords: Minimum classification error training; Discriminative training; Large-margin training; Speech recognition; Large-scale speech recognition; Theoretical classification risk minimization

☆ Part of the work has been presented in "Use of incrementally regulated discriminative margins in MCE training for speech recognition" at Interspeech 2006, Lisbon, Portugal (Yu et al., 2006) and "Large-margin minimum classification error training for large-scale speech recognition tasks" at ICASSP 2007, HI, USA (Yu et al., 2007).

* Corresponding author. Tel.: +1 425 707 9282; fax: +1 425 706 7329.
E-mail addresses: dongyu@microsoft.com (D. Yu), deng@microsoft.com (L. Deng), xiaohe@microsoft.com (X. He), alexac@microsoft.com (A. Acero).

0885-2308/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2008.03.002

1. Introduction

Discriminative training for hidden Markov models (HMMs) has become one of the core technologies in speech recognition research over the past dozen years or so (Juang et al., 1997; Juang and Katagiri, 1992; McDermott, 1997; McDermott et al., 2007; Povey et al., 2004; Woodland and Povey, 2000). The central idea of most of the discriminative training algorithms is the optimization of empirical error rates in the training set either directly or indirectly, in the hope of improving the overall recognition accuracy.

Generalization ability is one of the key issues in discriminative training since the gains in the training set may not be translated to the gains in the test set. In the past, the improvement of the generalization ability was usually achieved by optimizing the smoothed empirical training set error rate. It was shown by McDermott and Katagiri (2002) that optimizing the smoothed error rate in minimum classification error (MCE) training is equivalent to optimizing the overall error rate of the whole observation space. Recently, much progress has been made to further improve the generalization ability by incorporating margins (i.e., the distance between the well classified samples and the decision boundary) into the discriminative training process (Li et al., 2006, 2007; Li and Jiang, 2005, 2006; Liu et al., 2006; Sha and Saul, 2006a,b, Jiang and Li, 2007). One such approach is to maximize the margins directly using the gradient descent (Li and Jiang, 2005; Liu et al., 2006) or semi-definite programming (Li and Jiang, 2005) when the training set error rate is very low. An alternative method is to optimize some form of combined scores of the margin and the empirical error rate (Li et al., 2006, 2007; Sha and Saul, 2006a,b; Jiang and Li, 2007).

A natural and interesting question is whether the existing discriminative training algorithms can be extended directly to embed the concept of margin. In this paper, we give this question an affirmative answer by showing that the sigmoid bias in the conventional minimum classification error (MCE) training can be interpreted as a soft margin. Following McDermott and Katagiri (2002, 2004), we justify this claim from a theoretical classification risk (sometimes also called Bayes risk) minimization perspective, showing that the loss function associated with a non-zero sigmoid bias includes not only empirical error rates but also a margin-bound risk so that some already correctly classified training samples can be further pushed away from the decision boundary.

Choosing the appropriate margin in one step, however, is not easy. In this paper we propose a practical optimization strategy that increases the margin (the sigmoid bias) incrementally over epochs in the MCE training process so that a desirable balance between the empirical error rates on the training set and the margin can be achieved and verified by cross validation. We show that adjusting the margin this way can reduce the number of non-outlier training utterances which would be treated as outliers (to be discussed in detail in Section 4). We call this modified MCE training process large-margin minimum classification error (LM-MCE) training to differentiate it from conventional MCE. Since LM-MCE is a direct extension to the MCE training it can be easily applied to the existing MCE systems.

The rest of the paper is organized as follows: In Section 2, we formulate the MCE training method as a theoretical classification risk minimization problem in both the feature domain and the (misclassification-measure) score domain using the Parzen-window distribution estimation method. In Section 3, we extend the results in Section 2 and show that the sigmoid bias in MCE can be interpreted as some form of margin from the theoretical classification risk minimization perspective with a loss function that includes both the empirical error rates and a margin-bound risk. In Section 4, we propose a practical optimization strategy for selecting the margin (sigmoid bias) in LM-MCE training. In Section 5, we apply LM-MCE to the TIDIGITS corpus (a small-scale speech recognition task) and to the Microsoft-internal large vocabulary telephony speech recognition database, and demonstrate that LM-MCE is effective for both the small-scale and the large-scale modeling and recognition tasks. We conclude the paper in Section 6.

2. Theoretical classification risk minimization and minimum classification error training

2.1. Minimum classification error training

MCE training (Juang et al., 1997; Juang and Katagiri, 1992; McDermott, 1997) was traditionally formulated as a problem of optimizing the smoothed empirical training-set sentence- or string-level error rate.

Assume that there are R observation samples (sentences or utterances) $\mathbf{X}_r$ ($r = 1, \ldots, R$) in the training set and that each sample $\mathbf{X}_r$ is a sequence of T_r vector-valued observation data (typically consisting of cepstra and their derivatives) $\mathbf{x}_{r,t}$ ($t = 1, \ldots, T_r$). Each sample $\mathbf{X}_r$ is associated with a reference label (e.g., a word sequence) $s_r \in \mathbb{C} = \{c_i | i = 1, \ldots, C\}$, where C denotes the total number of possible labels in the task. It should be noted that, for continuous ASR, C is the number of all possible permutations of words in the vocabulary constrained by the grammar and is usually infinite or very large. Therefore, often only the N most confusable competing strings are considered in MCE training (although recent work extended the N-best-list based training to lattice-based training (Macherey et al., 2005)).

Using the above notations, the ASR task can be considered as a C-class classification problem, where each observation sample $\mathbf{X}$ is to be classified into one of the C classes. The goal of ASR model training is to design a mapping or decision function $F(\mathbf{X})$ from the observation space $\mathbf{X} \in \aleph$ to the discrete set $c = F(\mathbf{X}) \in \mathbb{C}$, where the decision rule is

$$F(\mathbf{X}) = c_i \quad \text{iff } g_k(\mathbf{X}; \Lambda) - g_i(\mathbf{X}; \Lambda) < 0 \quad \forall k \neq i. \tag{1}$$

In (1), Λ represents the model parameters and $g_i(\mathbf{X}; \Lambda) = \log p(\mathbf{X}, c_i | \Lambda)$ is the discriminant function for class-i.

During the training phase, a misclassification measure $D_i(\mathbf{X}; \Lambda)$ is usually defined as

$$D_i(\mathbf{X}; \Lambda) \triangleq G_i(\mathbf{X}; \Lambda) - g_i(\mathbf{X}; \Lambda), \tag{2}$$

where $G_i(\mathbf{X}; \Lambda)$ is the anti-discriminant function. For the popular one-best MCE training when only the top-one incorrectly recognized string is used as the "most competitive candidate" for discriminative training,

$$G_i(\mathbf{X}; \Lambda) = \max_{k \neq i} g_k(\mathbf{X}; \Lambda) \tag{3}$$

and thus the decision rule (1) can be rewritten as

$$F(\mathbf{X}) = c_i \quad \text{iff } D_i(\mathbf{X}; \Lambda) \triangleq G_i(\mathbf{X}; \Lambda) - g_i(\mathbf{X}; \Lambda) < 0. \tag{4}$$

Note that the above "most competitive candidate" may change dynamically from iteration to iteration. In practice, it is generated through new decoding based on the model parameters obtained at the immediately previous iteration.

For the more general N-best MCE training where top $N > 1$ incorrectly recognized strings are used as the "competitive candidates", one practically useful way of constructing the anti-discriminant function is to use the soft-max function to approximate the max function and (3) becomes

$$G_i(\mathbf{X}; \Lambda) = \log \sum_{1 \leqslant k \leqslant N, n_k \neq i} \exp[g_{n_k}(\mathbf{X}; \Lambda)]. \tag{5}$$

Note that the soft-max function is differentiable with respect to the model parameters while the max function is not. Approximating the max function with the soft-max function thus allows for the use of gradient-based optimization algorithms (Katagiri et al., 1998). The goal of the MCE training is to minimize the total empirical risk

$$L_{\text{MCE}}(\Lambda) = \frac{1}{R} \sum_{r=1}^{R} l_r \tag{6}$$

in the training set, where l_r is the risk associated with the rth sample in the training set. A commonly used empirical risk is the zero-one risk function

$$l_r = \delta(D_{s_r}(\mathbf{X}_r, \Lambda) \geqslant 0) = \begin{cases} 0 & D_{s_r}(\mathbf{X}_r, \Lambda) < 0, \\ 1 & \text{elsewhere}, \end{cases} \tag{7}$$

where S_r is the reference label. The zero-one risk function (7) indicates that the risk associated with the correct classification is zero and that of misclassification is one. To make the loss function differentiable with respect to the model parameters, however, MCE training needs to optimize a smoothed version of the above zero-one risk function. The most popular smooth function used in MCE training is the sigmoid function

$$l_r = \frac{1}{1 + \exp(-\alpha D_{s_r}(\mathbf{X}_r, \Lambda) + \beta)}, \tag{8}$$

which is popular historically because it allows easy computation of the derivative from the function evaluation. In most cases, β in (8) is set to zero in the MCE training and (8) is simplified to

$$l_r = \frac{1}{1+\exp(-\alpha D_{s_r}(\mathbf{X}_r, \Lambda))}. \tag{9}$$

When $\beta = 0$ and (3) is used, i.e., the top-one candidate $s_{r,e}$ is used as the "competitive candidate" for discriminative training, it can be proven that (9) is equivalent to

$$l_r = \frac{p^{\alpha}(\mathbf{X}_r, s_{r,e}|\Lambda)}{p^{\alpha}(\mathbf{X}_r, s_{r,e}|\Lambda) + p^{\alpha}(\mathbf{X}_r, s_r|\Lambda)}. \tag{10}$$

This is the risk function used in our current implementation. Note that minimizing (6) with the risk function defined in (10) is equivalent to maximizing the MCE objective function

$$O_{\mathrm{MCE}}(\Lambda) = R(1 - L_{\mathrm{MCE}}(\Lambda)) = \sum_{r=1}^{R} \frac{p^{\alpha}(\mathbf{X}_r, s_r|\Lambda)}{p^{\alpha}(\mathbf{X}_r, s_{r,e}|\Lambda) + p^{\alpha}(\mathbf{X}_r, s_r|\Lambda)}. \tag{11}$$

2.2. Theoretical classification risk minimization view of MCE

McDermott and Katagiri (2002, 2004) and McDermott et al. (2007) showed that optimizing the MCE training criteria can be made equivalent to optimizing the estimated empirical error rate over the entire observation space using the Parzen-window-based non-parametric distribution estimation.

Given the classification scenario specified in Section 2.1, the expected overall risk over the entire observation (feature domain) space using the one-zero risk function (7) becomes

$$\begin{aligned}\Re &= \int_x [\bar{r}(F(\mathbf{X})|\mathbf{X})]p(\mathbf{X})\mathrm{d}\mathbf{X} = \int_x \left[\sum_{j=1}^{C} \delta[D_j(\mathbf{X};\Lambda) \geqslant 0]P(c_j|\mathbf{X})\right] p(\mathbf{X})\mathrm{d}\mathbf{X} \\ &= \sum_{j=1}^{C} P(c_j) \int_x \delta[D_j(\mathbf{X};\Lambda) \geqslant 0] p_x(\mathbf{X}|c_j)\mathrm{d}\mathbf{X} = \sum_{j=1}^{C} P(c_j) \int_{D_j(x;\Lambda)\geqslant 0} p_x(\mathbf{X}|c_j)\mathrm{d}\mathbf{X},\end{aligned} \tag{12}$$

where $\bar{r}$ is the expected risk for a fixed observation vector $\mathbf{X}$ and $P(c_j)$ is the prior class probability that can be estimated as $R_j/\sum_{i=1}^{C} R_i = R_j/R$ (R_j is the number of training samples for class c_j). We now convert the problem from the feature domain to the score domain noting that both the observation vector $\mathbf{X}$ and the misclassification measure $s_j = D_j(\mathbf{X};\Lambda)$ are random variables. Using the cumulative distribution technique for finding distributions of functions of random variables, we can express the integral for each category c_j over the space $D_j(x;\Lambda) \geqslant 0$ with an integral over the positive domain of the misclassification measure s_j. The expected classification risk (12) becomes

$$\begin{aligned}\Re &= \sum_{j=1}^{C} P(c_j) \int_{D_j(x;\Lambda)\geqslant 0} p_x(\mathbf{X}|c_j)\mathrm{d}\mathbf{X} = \sum_{j=1}^{C} P(c_j) P[D_j(x;\Lambda) \geqslant 0] = \sum_{j=1}^{C} P(c_j) P[s_j \geqslant 0] \\ &= \sum_{j=1}^{C} P(c_j) \int_0^{\infty} p_{D_j}(s_j|c_j)\mathrm{d}s_j = \sum_{j=1}^{C} P(c_j) \int_0^{\infty} p_{D_j}(D|c_j)\mathrm{d}D,\end{aligned} \tag{13}$$

where $p_{D_j}(D|c_j)$ is the distribution of the misclassification score for class c_j and many points in the feature domain may map to the same point in the score domain. $p_{D_j}(D|c_j)$ can be estimated using the Parzen window on the score domain as

$$p_{D_j}(D|c_j) \approx \frac{1}{R_j} \sum_{r=1}^{R_j} \frac{1}{H_r} W_D\left(\frac{D - D_{r,j}}{H_r}\right), \tag{14}$$

where $D_{r,j}$ is the misclassification score associated with the training data sample $\mathbf{X}_r$ labeled as class c_j, and H_r is the bandwidth of the one-dimensional kernel function W_D in the score domain.

It can be shown that using the sigmoid smoothing function in the MCE training is equivalent to using the symmetric kernel function

$$W_D(D) = \frac{1}{\left[\exp\left(-\frac{D}{2}\right) + \exp\left(\frac{D}{2}\right)\right]^2}. \tag{15}$$

This is due to the fact that given (15), we have

$$\begin{aligned}\Re &= \sum_{j=1}^{C} P(c_j) \int_0^{\infty} \frac{1}{R_j} \sum_{r=1}^{R_j} \frac{1}{H_r} W_D\left(\frac{D - D_{r,j}}{H_r}\right) \mathrm{d}D = \sum_{j=1}^{C} \frac{P(c_j)}{R_j} \sum_{r=1}^{R_j} \int_{\frac{-D_{r,j}}{H_r}}^{\infty} W_D(U)\mathrm{d}U \\ &= \sum_{j=1}^{C} \frac{P(c_j)}{R_j} \sum_{r=1}^{R_j} \int_{\frac{-D_{r,j}}{H_r}}^{\infty} \frac{1}{\left[\exp\left(-\frac{U}{2}\right) + \exp\left(\frac{U}{2}\right)\right]^2} \mathrm{d}U = \sum_{j=1}^{C} \frac{P(c_j)}{R_j} \sum_{r=1}^{R_j} \frac{1}{1 + \exp(-D_{r,j}/H_r)} \\ &= \frac{1}{R} \sum_{j=1}^{C} \sum_{r=1}^{R_j} \frac{1}{1 + \exp(-D_{r,j}(x_r; \Lambda)/H_r)}.\end{aligned} \tag{16}$$

This is equivalent to (8) (the conventional MCE training criteria with sigmoid loss function) if

$$\alpha = \frac{1}{H_r} \tag{17}$$

and

$$\beta = 0. \tag{18}$$

We want to draw attention to several observations here. First, we can justify the conventional MCE training using the theoretical classification risk minimization framework and the Parzen-window-based non-parametric distribution estimation. Second, the sigmoid function is just one of the loss functions that can be used in the MCE training. Many different loss functions can be derived by choosing different kernels. Third, with smoothing, conventional MCE optimizes an estimated empirical error rate on the entire observation space (instead of just the training set) if the training set is representative. Since the probability distribution of the entire observation space is estimated using the Parzen-window kernel method, the variance of the estimation is bounded and the estimation converges to the true empirical error rate as the training set size increases. This suggests that conventional MCE has some built-in generalization ability. In fact, this property becomes apparent after examining the following property of the sigmoid loss function: if a token is correctly classified but is close to the decision boundary, the cost associated with this token is still greater than zero. This near-miss situation reflects the possibility that a similar (but not identical) token in the test set might be misclassified. On the other hand, a token that is mis-classified in the training set and is close to the decision boundary would have a cost less than one, indicating that a similar token in the test set might be correctly classified. Fourth, the bandwidth H_r (hence α) can be dependent on each training sample (e.g., estimated based on the k-nearest neighbors of the sample) and is not necessarily fixed for the entire training set as in the traditional MCE training (Juang et al., 1997).

3. Large-margin minimum classification error training

In this section, we show that the generalization ability of the MCE training with $\beta = 0$ can be further improved through LM-MCE, which embeds discriminative margins in the margin-free theoretical classification risk of (13). To make the discussion easier and more concrete, we define the discriminative margin in the score space as a non-negative value $m \geqslant 0$, which represents the extent of the classifier's tolerance gap, or the value in the score space over which the correct classification near the decision boundary is deemed incorrect nevertheless.

We now modify the margin-free version of the integration space in (12): $\{x\colon D_j(x;\Lambda) \geqslant 0\}$ to the new, margin-sensitive one: $\{x\colon D_j(x;\Lambda) \geqslant -m\}$. As a result, (13) is changed to

$$\begin{aligned}\Re &= \sum_{j=1}^{C} P(c_j) \int_{-m}^{\infty} p_{D_j}(D|c_j)\mathrm{d}D \\ &= \underbrace{\sum_{j=1}^{C} P(c_j) \int_{0}^{\infty} p_{D_j}(D|c_j)\mathrm{d}D}_{\text{margin-free Bayes risk}} + \underbrace{\sum_{j=1}^{C} P(c_j) \int_{-m}^{0} p_{D_j}(D|c_j)\mathrm{d}D}_{\text{margin-bound Bayes risk}}.\end{aligned} \tag{19}$$

Note that, there are two parts in this new risk function, a margin-free part which equals to the risk function we have discussed in Section 2 and a margin-bound part whose value depends on the margin m. Expressed in another way, this new risk function is a combination of the estimated empirical error rate and a function of the pre-defined margin m. This pre-defined margin can be considered as a required margin whose value can be tuned on a held-out set. Also note that although it looks to be quite different from other large-margin techniques proposed (Sha and Saul, 2006a,b, Jiang and Li, 2007), they are similar in spirit since all these methods aim to optimize some function of the empirical error rate and the margin. A detailed analysis and comparison can be found in (Yu and Deng, 2007).

With this new risk defined in (19), (16) is accordingly changed to

$$\begin{aligned}\Re &= \sum_{j=1}^{C} P(c_j) \int_{-m}^{\infty} \frac{1}{R_j} \sum_{r=1}^{R_j} \frac{1}{H_r} W_D\left(\frac{D - D_{r,j}}{H_r}\right)\mathrm{d}D = \sum_{j=1}^{C} \frac{P(c_j)}{R_j} \sum_{r=1}^{R_j} \int_{\frac{-m-D_{r,j}}{H_r}}^{\infty} W_D(U)\mathrm{d}U \\ &= \sum_{j=1}^{C} \frac{P(c_j)}{R_j} \sum_{r=1}^{R_j} \frac{-1}{(1+\exp U)}\Bigg|_{\frac{-m-D_{r,j}}{H_r}}^{\infty} = \sum_{j=1}^{C} \frac{P(c_j)}{R_j} \sum_{r=1}^{R_j} \frac{1}{1+\exp[-(D_{r,j}+m)/H_r]} \\ &= \frac{1}{R} \sum_{j=1}^{C} \sum_{r=1}^{R_j} \frac{1}{1+\exp[-(D_{r,j}(x_r;\Lambda)+m)/H_r]}.\end{aligned} \tag{20}$$

The motivation for the above decomposition of the total risk into margin-free and margin-bound components is to demonstrate that the intuitive concept of margin introduced earlier is indeed related to the theoretically motivated classification risk. Let us discuss further insights from the above margin-sensitive theoretical classification risk where the lower limit of the integration is $-m$ instead of zero as in the conventional margin-free theoretical classification risk. One interesting aspect is the relation between margin, Parzen-window width, and the training-set size. As with the Parzen-window width, one natural approach is to reduce m as the training-size increases (i.e., the generalization ability becomes less of an issue with the increased training size), so that the summation of all windows induced by the training set covers the entire misclassification-measure space with as little overlap and as little bias as possible. If we choose the margin and Parzen-window width in this way, then we have the convergence of our proposed margin-sensitive theoretical classification risk of (19) to the true risk as the sample size goes to infinity. The convergence is achieved because both the Parzen-window width and the margin will go to zero in the limit.

Comparing with (8), we see that (20) equals to the conventional MCE training criterion with the following token-adaptive relationship

$$\alpha = \frac{1}{H_r} \tag{21}$$

and

$$\beta = \frac{-m}{H_r}. \tag{22}$$

The margin-sensitive theoretical classification risk in the form of (20) can be viewed as a principled extension to and a special flavor of the conventional MCE in that a non-zero valued discriminative margin is introduced to improve the gap tolerance and generalization ability of the classifier. Note that although some of the conventional MCE training implementations also used a non-zero β, their purpose is to fit the distribution of the training data instead of increasing the margin (e.g., McDermott, 1997). Our current work distinguishes

itself from the earlier one in terms of the use of non-zero β not only in conceptual ways, but also in practical ways (See Section 4). Further, with the formulation of (20), our LM-MCE concept can be easily extended to other kernel functions.

4. Optimization strategy

MCE training was carried out in many studies using the generalized probabilistic descent (GPD) algorithm (e.g., Juang et al., 1997; Juang and Katagiri, 1992; McDermott, 1997; Jiang et al., 2002) or Extended Baum–Welch algorithm (Macherey et al., 2005), although most of the large vocabulary MCE work has mainly used gradient descent of some form (e.g., Le Roux and McDermott, 2005; Schlueter et al., 2001; McDermott et al., 2000). In this work, we have implemented the MCE algorithm that maximizes (11) by a special technique of optimization via growth transformation. This implementation is an improvement upon the implementation that was originally proposed by He and Chou (2003). The improvement lies in converting the super-string-level objective function into a normal string-level objective function for MCE. This conversion is accomplished via a non-trivial mathematical framework (He et al., 2006), which results in a rational function that is then subject to optimization by growth transformation or extended Baum–Welch algorithm. Using this new growth-transformation-based optimization, many fewer iterations are required for empirical convergence than those typically required by the gradient-based GPD (Juang et al., 1997).

Unlike the conventional MCE training, the motivation of the LM-MCE training is to improve the generalization ability of the conventional MCE training by optimizing the combined risk defined in (19). In this optimization problem, one of the key parameters to determine is margin variable m. Choosing an appropriate margin in one step, however, is not easy. Following the method used in support vector machines in determining the appropriate balance between the empirical error rate (margin-free risk) and the margin, we propose increasing the margin gradually over epochs and selecting the optimal margin through cross verification. In other words, the margin is originally set to 0 or even negative (to incorporate some utterances that are treated as outliers when $m = 0$ as will be discussed in the next paragraph). Then, the margin is increased gradually over epochs. The training process (as well as the change of the margin) stops when the minimum word error rate (WER) on the development set is achieved. As will be discussed shortly, this strategy carries with it an additional benefit in training the model parameters.

The MCE training criterion has the property of immunity to the outliers (e.g., the utterances that are grossly mis-recognized as a result of mislabeling). This is due to the fact that the most an utterance can contribute to the loss function (error rate) is one. We can examine this property from another perspective. Both the GPD and the growth-transformation algorithms update the HMM parameters using the derivatives of the loss function. If the score of an utterance is far away from the center of the sigmoid function (where the sigmoid function saturates), the derivative associated with this utterance will become close to zero, and thus this utterance will not contribute to updating of the HMM parameters. However, this may also cause some non-outlier utterances be treated as outliers during the training process, leading to a less optimal model. This drawback can be alleviated using techniques similar to simulated annealing, where a small α is chosen initially to include as many utterances as possible in updating the HMM parameters. α is then gradually increased to a point that a good approximation of the test set error rate can be achieved based on (16).

The introduction of the margin does not change the basic parameter updating algorithms. However, setting a fixed large margin in one step may cause the training algorithm to treat additional utterances as outliers (as illustrated in Fig. 1) and thus hurt the training performance, especially if α is also fixed. We use Fig. 1 to illustrate this, where the utterances represented by circles are assumed to belong to class 1 and those represented by triangles to class 2. In the upper sub-figures, margins are set to zero while in the lower sub-figures margins are set to a positive value. As can be seen, the utterance represented by the right-most circle in the upper-left sub-figure is not treated as an outlier utterance. However, when the margin is set to a fixed large value, it is treated as an outlier as indicated in the lower-left sub-figure. This tends to lead to a sub-optimal solution. Note that this drawback can be alleviated with our proposed optimization strategy of gradually increasing margins over epochs. Implementation details of this strategy will be presented in the next section.

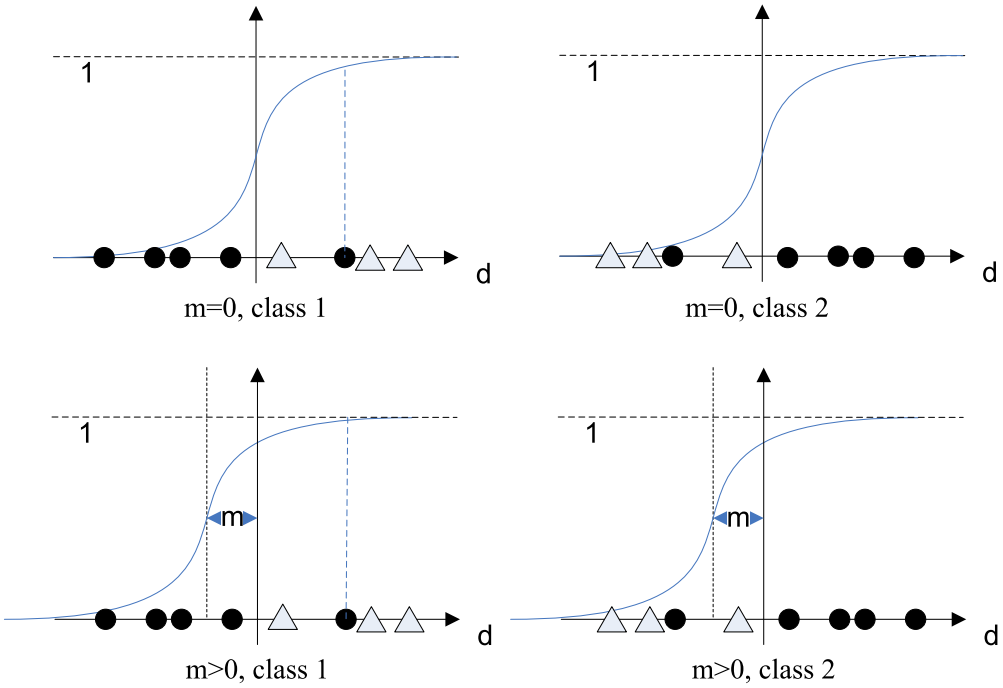

Fig. 1. Illustration of LM-MCE.

5. Experiments

We have conducted experiments on two corpora: the TIDIGITS database and the Microsoft telephony large-scale speech recognition database.

5.1. Experimental results on TIDIGITS

TIDIGITS corpus (Leonard, 1984) contains utterances from 326 speakers (111 men, 114 women, and 101 children) from 21 regions of the United States. The vocabulary size of the database is 11 – digits of "1" to "9", plus "oh" and "zero". Each utterance in the database is a connected-digit string whose length varies between one and seven (except there are no six-digit strings). In our experiments, we only use the adult portion of the database, which makes up a standard training set of 8623 digit strings (from 55 men and 57 women) and a standard test set of 8700 digit strings (from 56 men and 57 women).

In our experiments, all data are sampled at a rate of 16 K Hz. The 33-dimensional acoustic feature vectors are composed of the normalized energy, 10 MFCCs (Mel-Frequency Cepstrum Coefficients), and their first and second order time derivatives. HMMs are built for each of the ten digits from *ZERO* to *NINE*, plus word *OH*. We use head–body–tail CDHMMs where each digit word is split into three parts and each part is modeled with different HMMs. The body, which is the middle part of the word, is assumed to be context-independent. The head and the tail are dependent on the previous and subsequent digit (or silence), respectively. In our experiments the head and tail models consisted of three states, whereas the number of states in body models varies depending on the mean duration of the digit. We use a different number of Gaussian mixture components for each state in our experiments. The total number of Gaussian mixture components used in the system is 3284, which is roughly the same as the number in a nine-state whole-word CDHMMs with 32 Gaussians per state. The Baum–Welch re-estimation algorithm in the standard HTK is used to train the baseline system.

Our LM-MCE training algorithms (with and without the discriminative margin) are applied after the HMMs are initialized using the Maximum Likelihood (ML) training criterion. The word error rate (WER) and string error rate (SER) in the test set using the initial ML-trained models are 0.28% and 0.78%, respectively, with the tuned insertion penalty of −14.5 and language model weight of −13.25. The WER of 0.23%

and SER of 0.68% are the best MCE baselines (i.e., no discriminative margin is used) we had obtained on this task. They are achieved when the Parzen-window bandwidth is tuned to $H_r = 120$ (i.e., when α in the sigmoid function is tuned to 1/120). This represents 17.86% relative WER reduction and 12.82% relative SER reduction over the initial ML-trained models. In all MCE experiments, we use N-best lists (when $N = 20$) to represent the digit-string competitors. (We have not used lattices for the competitor representation, which may give superior results as demonstrated in Schlueter et al. (2001).)

To show the difference before and after the incrementally regulated discriminative margin is applied, we keep $H_r = 120$ (or $\alpha = 1/120$) and use three different methods for setting the margin values. The three methods are tested under otherwise identical experimental conditions.

In the first method, the margin $m(i)$ (hence $\beta(i)$ in the sigmoid function) is set to be independent of the epoch number with a value over the range of [0, 120] (or [−1, 0] for $\beta(i)$). The initial HMMs for MCE training with each of the fixed m values are from ML training. A total of 15 MCE growth-transformation iterations are used for each of the fixed margin values.

In the second method, the margin $m(i)$ changes from neutral (no margin or $m = 0$) to $m = 120$, with a step size of 12 (heuristically determined based on H_r and the distribution of the training set misclassification measure), during the LM-MCE training. That is, $m(i) = 12(i - 1)$ (or $\beta(i) = -0.1(i - 1)$) for $i = 1, \ldots, 11$. When $m = 0$ (and so $\beta = 0$) the HMMs are trained using the LM-MCE algorithm (four iterations) from the ML-trained models. As the margin incrementally increases, the previously LM-MCE-trained models serve as the base for additional four LM-MCE iterations using a new m (and β) value.

In the third method, the margin $m(i)$ changes from −48 to 60, with a step size of 12 also. That is, $m(i) = 12(i - 1) - 48$ (or $\beta(i) = 0.4 - 0.1\ (i - 1)$) for $i = 1, \ldots, 10$.

Figs. 2–4 depict the error rates (WER and SER) for both the training set and the test set as a function of the epoch number for methods 1–3, respectively. The effects of increasing the discriminative margin upon the recognition errors can be observed from these figures. Errors tend to reduce in the beginning when the margin enlarges and then to increase as the margin further increases. In other words, the largest margin does not correspond to the lowest error. The figures also reveal that the lowest training and test error rates do not occur at the same m value.

The overall experimental results using the three methods discussed above are summarized in Table 1. Note that traditionally there is no development set in TIDIGTS corpus. It is also difficult to reserve a development set from the training set since the error rate in the training set is extremely low for this task (with $\leqslant$0.14%WER). For this reason, we report two error rate figures in Table 1 for each margin-setting method: one is the best we can obtain (which is the number usually reported in the literature) and the other is the result

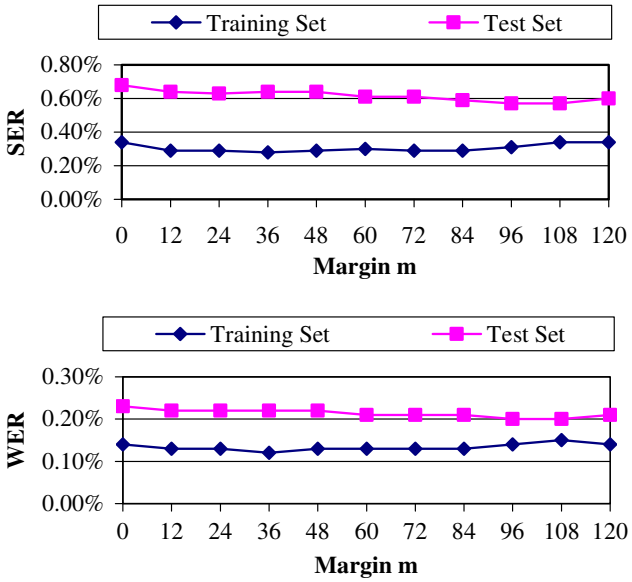

Fig. 2. Recognition error rate as a function of margin m, which is fixed over MCE training iterations (Method 1). Note that in this case there is no direct relationship between models with different margins.

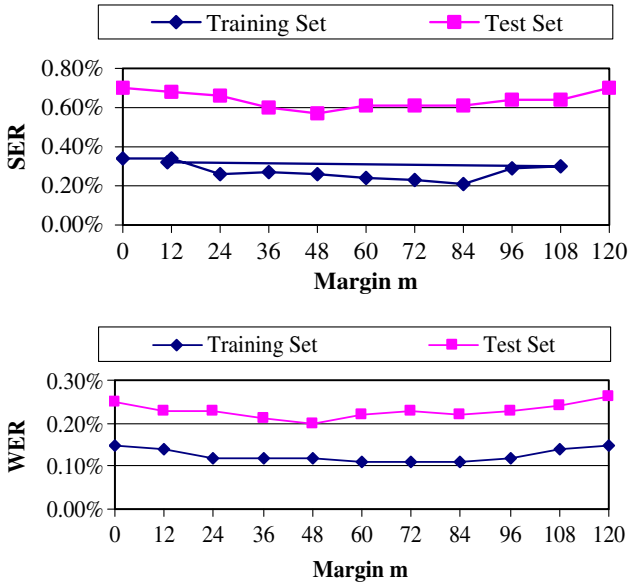

Fig. 3. Recognition error rate as a function of margin m, which is increased over MCE training iterations from 0 to 120 with an increment of 12 (Method 2).

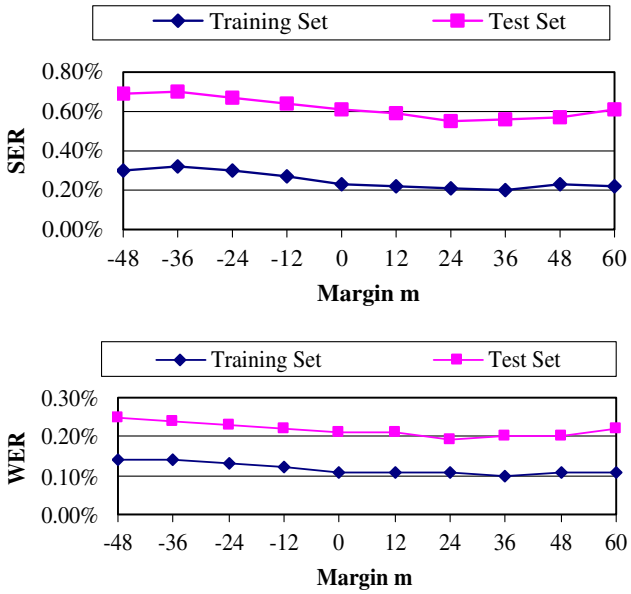

Fig. 4. Recognition error rate as a function of margin m, which is increased over MCE training iterations from −48 to 60 with an increment of 12 (Method 3).

Table 1
Summary of the experimental results on TIDIGITS (columns labeled with "Relative" refer to relative error rate reduction)

Margin	WER (lower)		WER (upper)		SER (lower)		SER (upper)	
	Absolute (%)	Relative (%)	Absolute (%)	Relative (%)	Absolute (%)	Relative (%)	Absolute (%)	Relative (%)
$m = 0$	0.23	Baseline	0.23	Baseline	0.68	Baseline	0.68	Baseline
Method 1	0.20	13.04	0.22	4.35	0.57	16.18	0.64	5.88
Method 2	0.20	13.04	0.22	4.35	0.57	16.18	0.61	10.29
Method 3	*0.19*	17.39	*0.20*	13.04	*0.55*	19.12	*0.56*	17.65

we obtain when the training-set error reaches minimum (i.e., using the training set as the development set). If a true development set were available, the result would be somewhere between these two numbers. In other words, we regard these performance numbers as lower and upper WER/SER, respectively.

In Table 1, relative error reduction is calculated compared with the MCE baseline where the discriminative margin m is set to zero. We observe 13.04% relative WER reduction and 16.18% relative SER reduction over the baseline MCE models with Methods 1 and 2 in the best case. By using Method 3, we have achieved 0.19% absolute WER and 0.55% absolute SER, which translate to 17.39% relative WER and 19.12% relative SER reduction over our MCE baseline in the best case, and 13.04% relative WER and 17.65% relative SER reduction when the error rates in the training set reaches its minimum. The gain has been tested to be statistically significant at the significance level of 5% with McNemar's test on the sentence error rate.

Also noticeable from Table 1 is that the gap between the lower and the upper error rates has the minimum value with Method 3. This indicates that the gain in the training set is well translated to the test set in Method 3. This can also be confirmed by the consistent trend in Fig. 4.

We also have investigated the effect of the Parzen-window bandwidth H_r upon the LM-MCE training result. Due to the use of the sentence-level discriminant function in the LM-MCE, it has not been possible to collect reliable statistics for estimating the adaptive kernel bandwidth H_r for each sample due to the lack of sufficient samples. In this study we only vary the value of H_r in different experimental settings but we fix it for all training samples. Table 2 compares the results between $H_r = 30$ and $H_r = 120$ using Method 1 described above. Table 2 confirms the benefit of incorporating margins in the discriminative training criterion. In particular, we see large error reduction by using non-zero margins with these two different bandwidths. We also observe that with an increasing margin value, the difference between the performances of the training and test data is decreasing; e.g., the SER difference of 0.51% when $m = 0$ is reduced to 0.26% when $m = 120$. This provides evidence that the use of margins contributes to reducing the training-test data mismatch due to sampling. Finally, from Table 2, we observe the effect of the Parzen-window bandwidth as follows: LM-MCE training with an appropriate Parzen-window bandwidth tends to improve the recognition accuracy over other bandwidths.

5.2. Experimental results on microsoft large-scale telephony speech database

The Microsoft large-scale telephony speech databases are used to build a large vocabulary telephony ASR system. The entire training set, which is collected through various channels including close-talk telephones, far-field microphones, and cell phones, consists of 26 separate corpora, 2.7 million utterances, and a total

Table 2
Speech recognition error rates for the training and test sets as a function of the fixed margin and kernel-bandwidth parameters in theoretical classification risk of Eq. (20)

Margin (m)	$Hr = 30$				$Hr = 120$			
	Training		Test		Training		Test	
	SER%	WER%	SER%	WER%	SER%	WER%	SER%	WER%
0	0.23	0.11	0.74	0.26	0.34	0.14	0.68	0.23
12	0.23	0.11	0.74	0.26	0.29	0.13	0.64	0.22
24	0.26	0.12	0.71	0.25	0.29	0.13	0.63	0.22
36	0.28	0.13	0.70	0.25	0.28	0.12	0.64	0.22
48	0.28	0.13	0.69	0.25	0.29	0.13	0.64	0.22
60	0.28	0.12	0.67	0.24	0.30	0.13	0.61	0.21
72	0.29	0.13	0.68	0.24	0.29	0.13	0.61	0.21
84	0.29	0.13	0.66	0.23	0.29	0.13	0.59	0.21
96	0.32	0.14	0.64	0.23	0.31	0.14	0.57	0.20
108	0.32	0.13	0.63	0.23	0.34	0.15	0.57	0.20
120	0.35	0.14	0.61	0.22	0.34	0.14	0.60	0.21
132	0.36	0.15	0.64	0.23				
144	0.38	0.16	0.67	0.24				
156	0.38	0.16	0.68	0.24				
168	0.41	0.16	0.70	0.24				

The recognizer's parameters (HMM mean vectors and covariance matrices) are learned by minimizing margin-sensitive theoretical classification risk.

Table 3
Description of the test sets

Name	Voc size (K)	Word count	Audio (Hours)	Description
MSCT	70	4356	1	General call center application
STK	40	12851	3.5	Finance applications (stock transaction, etc.)
QSR	55	5718	1.5	Name dialing application (note: pronunciations of most names are generated by letter-to-sound rules)

of 2000 h of speech data. To improve the robustness of the acoustic model, speech data are recorded under various conditions with different environmental noises and include both native English speakers and speakers with various foreign accents. The text prompts include common telephony-application style utterances and some dictation-style utterances from the Wall Street Journal database.

The test sets consist of several typical context free grammar (CFG) based commercial telephony ASR tasks. To evaluate the generalization ability of our approach the test data are collected in a very different setup than the training set. The overall vocabulary size of the ASR system is 120 K. However, different vocabularies are used in different test sets. Table 3 summarizes the test sets used in our experiments, with the number of hours of audio data ranging from 1 to 3.5. Unlike the TIDIGITS case, in this task, a development set that is different from the test set is available for cross verification.

In this experiment, all data are sampled at a rate of 8 K Hz. Phonetic decision trees are used for state tying and there are about 6000 tied states with an average of 16 Gaussian mixture components per state. The 52-dimensional raw acoustic feature vectors are composed of the normalized energy, 12 MFCCs (Mel-Frequency Cepstrum Coefficients) and their first, second and third order time derivatives. The 52-dimensional raw features are further projected to form 36-dimensional feature vectors via heteroscedastic linear discriminant analysis (HLDA) transformation (Kumar and Andreou, 1998).

As with the TIDIGITS database, LM-MCE training is performed with initialization from the ML-trained models and with the window bandwidth H_r tuned to 30. In LM-MCE training of HMMs, the training data are first decoded by a simple unigram weighted CFG and the competitors are then updated after each set of three iterations. All HMM parameters (except transition probabilities) are updated. In order to prevent variance underflow, a dimension-dependent variance floor is set to be 1/20 of the average variance over all Gaussian components in that dimension. The variance values that are lower than the variance floor are set to the floor value.

Only two full sweeps of the entire training data set with two different margin values (i.e., two epochs) are used in LM-MCE training: the first epoch is performed with $m = 0$ and it takes three iterations. The second epoch is performed with $m = 6$ and it also takes three iterations. (We have tried one more epoch with $m = 12$ but did not observe further performance improvement in the development and the test set.) Better improvement might be achieved if the process was continued with decreased (usually by half) margin adjustment step. However, due to the high cost[1] of training on such a large database, we did not optimize the system further. The growth-transformation-based training algorithm (He et al., 2006) is used in the LM-MCE for fast convergence.

Table 4 presents the WER on the three test sets. Compared with the ML baseline, the conventional MCE training reduces the WER by 11.58% relatively. LM-MCE training further reduces the WER and achieves 16.57% relative WER reduction over the ML baseline across three test sets. In other words, LM-MCE training achieved 5.645% relative WER reduction over the conventional MCE even though only two epochs of the LM-MCE are carried out. The WER reduction over the conventional MCE is statistically significant at the significance level of 1% using the two-sided test. These results demonstrate that the LM-MCE training approach has strong generalization ability. It can be effectively applied not only to small-scale but also to large-scale ASR tasks. To the best knowledge of the authors, this is the first large-margin technique that

[1] We used 60 CPUs whose clock rates range from 1.8 GHz to 2.8 GHz. It takes 8 h to run one ML re-estimation iteration, 16 h to run one MCE re-estimation iteration, and 30 h to re-decode the training set and re-generate the competitors in each epoch. The total time required for each epoch is $30 + 16 \times 3 = 78$ h without taking into account disk I/O and network failure issues.

Table 4
Experimental results on the three telephony ASR test sets

	Test set	ML	MCE	LM-MCE (%)
MSCT	WER	12.413%	10.514%	10.009
	Abs. WERR Over ML	N/A	1.899%	2.404
	Rel. WERR Over ML	N/A	15.30%	19.37
	Abs. WERR Over MCE	N/A	N/A	0.505
	Rel. WERR Over MCE	N/A	N/A	4.803
STK	WER	7.993%	7.330%	6.926
	Abs. WERR Over ML	N/A	0.663%	1.067
	Rel. WERR Over ML	N/A	8.30%	13.35
	Abs. WERR Over MCE	N/A	N/A	0.404
	Rel. WERR Over MCE	N/A	N/A	5.512
QSR	WER	9.349%	8.464%	7.887
	Abs. WERR Over ML	N/A	0.885%	1.463
	Rel. WERR Over ML	N/A	9.47%	15.64
	Abs. WERR Over MCE	N/A	N/A	0.577
	Rel. WERR Over MCE	N/A	N/A	6.817
Average	WER	9.918%	8.769%	8.274
	Abs. WERR Over ML	N/A	1.149%	1.644
	Rel. WERR Over ML	N/A	11.58%	16.57
	Abs. WERR Over MCE	N/A	N/A	0.495
	Rel. WERR Over MCE	N/A	N/A	5.645

has been successfully applied to such a large-scale ASR task. We want to point out that the soft-margin estimation algorithm (Li et al., 2007) which has been successfully applied to the 5k-vocabulary Wall Street Journal task also has the potential to be successful when applied to our task.

6. Summary and conclusions

Use of large margins to improve the robustness and generalization performance of pattern recognition has been well motivated and is a standard practice for discriminative training in machine learning (Mason et al., 2000; Vapnik, 1998). Yet most practices of discriminative training in speech recognition have not embraced the concept of large margins and have been concerned mainly with empirical error rates in the training set (Juang et al., 1997; Juang and Katagiri, 1992; McDermott, 1997; McDermott et al., 2007; Povey et al., 2004; Woodland and Povey, 2000). The recent work of (Li et al., 2006; Li and Jiang, 2005, 2006; Liu et al., 2006; Sha and Saul, 2006a,b) introduced large margins in training HMMs for speech recognition, after the HMMs are pre-trained by the standard zero-margin MCE method. This paper reinterprets the sigmoid bias in the conventional MCE training as margin and reports an alternative optimization strategy where margins and empirical errors are jointly optimized in a generalized version of MCE. Superior recognition results on two ASR tasks, one large-scale and one small-scale, are obtained by our new, margin-sensitive method compared with the margin-free MCE method.

The idea behind our new method is the incorporation of incrementally adjusted margin, over the MCE training epochs, in the loss function of the MCE algorithm. In this way, empirical error rates and the discriminative margins are simultaneously optimized. The tradeoffs of introducing the new margin parameter are: (1) increased margins help generalization from the training set to the test set; (2) increased margins also create potential danger of sacrificing discrimination on the training set and of possibly sacrificing discrimination on the test set as well. To strike a balance between these two factors working against each other, we have developed a practical technique of incrementally regulating the change of the margin parameter over the MCE iterations. Experimental results show that the use of these incrementally changed margins significantly improves the prior art of margin-free MCE. In our current work, the sigmoid slope is fixed and inherited from the conventional MCE training. Adjusting the sigmoid slope using simulated-annealing techniques along with the adjustment of the sigmoid bias may lead to even better performance. We leave this as future work.

One important theoretical contribution of this work is the formulation of our LM-MCE training in terms of a theoretical classification risk minimization problem. From this perspective, we are able to generalize the conventional MCE not only by introducing the margin but also by making the margin parameter and the slope parameter adaptive to individual training tokens. In addition, it provides the possibility of selecting different types of Parzen-windows and hence different kinds of loss functions. We have not implemented algorithms related to these newly open possibilities in the work reported in this paper.

The main experimental contribution of this work is to successfully apply LM-MCE to train a large-scale speech recognition system. To our best knowledge, this is the first time the margin-based discriminative training is successfully applied to speech recognition tasks with a very large vocabulary size and with a massive amount of training data. In our experiments, LM-MCE has been extensively tested on multiple database-independent test sets covering a large number of commercial telephony ASR applications and conditions. The experimental results demonstrate that the LM-MCE not only works for small-vocabulary ASR tasks (such as TIDIGITS) but is also well suited for large-scale model training, and it can achieve significant performance improvement on both small-scale and large-scale ASR tasks.

Acknowledgement

We thank Dr. Hui Jiang of York University for useful discussions and the earlier large-margin training method which motivated this work.

References

He, X., Deng, L., Chou, W., 2006. A novel learning method for hidden Markov models in speech and audio processing. In: Proc. IEEE MMSP 2006.

He, X., Chou, W., 2003. Minimum classification error linear regression for acoustic model adaptation of continuous density HMMs. In: Proc. ICASSP.

Jiang, H., Siohan, O., Soong, F.-K., Lee, C.-H., 2002. A Dynamic In-search Discriminative Training Approach for Large Vocabulary Speech Recognition, vol. 1, pp. 113–116.

Jiang, H., Li, X., 2007. Incorporating training errors for large margin HMMs under semi-definite programming framework. In: Proc. ICASSP 2007, vol. IV, pp. 629–632.

Juang, B.-H., Chou, W., Lee, C.-H., 1997. Minimum classification error rate methods for speech recognition. IEEE Trans. Speech Audio Proc. 5 (3), 257–265.

Juang, B.-H., Katagiri, S., 1992. Discriminative training. ASJ Special Issue 3 (6), 333–339.

Katagiri, S., Juang, B.-H., Lee, C.-H., 1998. Pattern recognition using a generalized probabilistic descent method. Proc. IEEE 86 (11), 2345–2373.

Kumar, N., Andreou, A.G., 1998. Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition. Speech Commun. 26, 283–297.

Le Roux, J., McDermott, E., 2005. Optimization methods for discriminative training. In: Proc. Interspeech 2005, pp. 3341–3344.

Leonard, R.G., 1984. A database for speaker-independent digit recognition. In: Proc. ICASSP 1984, pp. 42.11.1–42.11.4.

Li, J., Yuan, M., Lee, C.-H., 2006. Soft margin estimation of hidden Markov model parameters. In: Proc. Interspeech 2006, pp. 2422–2425.

Li, J., Siniscalchi, S.M., Lee, C.-H., 2007. Approximate test risk minimization through soft margin estimation. In: Proc. ICASSP 2007, vol. IV, pp. 653–656.

Li, X., Jiang, H., 2005. A constrained joint optimization method for large-margin HMM estimation. In: Proc. ASRU Workshop, pp. 151–156.

Li, X., Jiang, H., 2006. Solving large-margin estimation of HMMs via semidefinite programming. In: Proc. Interspeech 2006, pp. 2414–2417.

Liu, C., Jiang, H., Rigazio, L., 2006. Recent improvement on maximum relative margin estimation of HMMs for speech recognition. In: Proc. ICASSP 2006, vol. 1, pp. 269–272.

Macherey, W., Haferkamp, L., Schlüter, R., Ney, H., 2005. Investigations on error minimizing training criteria for discriminative training in automatic speech recognition. In: Proc. INTERSPEECH-2005, pp. 2133–2136.

Mason, L., Bartlett, P., Baxter, J., 2000. Improved generalization through explicit optimization of margins. Mach. Learn. 38 (3), 243–255.

McDermott, E., 1997. Discriminative training for speech recognition. Ph.D. thesis, Waseda University, 1997.

McDermott, E., Biem, A., Tenpaku, S., Katagiri, S., 2000. Discriminative training for large vocabulary telephone-based name recognition. In: Proc. ICASSP, vol. 6, pp. 3739–3742.

McDermott, E., Katagiri, S., 2002. A Parzen window based derivation of minimum classification error from the theoretical Bayes classification risk. In: Proc. ICSLP.

McDermott, E., Katagiri, S., 2004. A derivation of minimum classification error from the theoretical classification risk using Parzen estimation. Comput. Speech Lang. 18, 107–122.

McDermott, E., Hazen, T., Le Roux, J., Nakamura, A., Katagiri, S., 2007. Discriminative training for large vocabulary speech recognition using minimum classification error. IEEE Trans. Speech Audio Proc. 15 (1), 203–223.

Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., and Zweig, G., 2004. fMPE: Discriminatively trained features for speech recognition. In: Proc. DARPA EARS RT-04 Workshop, 2004, Paper No. 35, p. 5.

Schlueter, R., Macherey, W., Muller, B., Ney, H., 2001. Comparison of discriminative training criteria and optimization methods for speech recognition. Speech Commun. 34 (3), 287–310.

Sha, F., Saul, L., 2006a. Large-margin Gaussian mixture modeling for phonetic classification and recognition. in: Proc. ICASSP 2006, vol. 1, pp. 265–268.

Sha, F., Saul, L., 2006b. Large-margin training of continuous-density hidden Markov models. in: Proc. NIPS, Vancouver, BC.

Vapnik, V., 1998. Statistical Learning Theory. Wiley.

Woodland, P.C., Povey, D., 2000. Large-scale discriminative training for speech recognition. In: Proc. ITRW ASR, ISCA, 2000, pp. 7–16.

Yu, D., Deng, L., He, X., Acero, A., 2006. Use of incrementally regulated discriminative margins in MCE training for speech recognition. In: Proc. Interspeech 2006, pp. 2418–2421.

Yu, D., Deng, L., He, X., Acero, A., 2007. Large-margin minimum classification error training for large-scale speech recognition tasks. In: Proc. ICASSP 2007, pp. 1137–1140.

Yu, D., Deng, L., 2007. Large-margin discriminative training of hidden markov models for speech recognition. In: Proc. ICSC 2007.