Approximate Test Risk Bound Minimization through Soft Margin Estimation

Jinyu Li, Ming Yuan, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—Inspired by the great success of margin-based classifiers, there is a trend to incorporate the margin concept into hidden Markov modeling for speech recognition. Several attempts based on margin maximization were proposed recently. In this paper, a new discriminative learning framework, called soft margin estimation (SME), is proposed for estimating the parameters of continuous density hidden Markov models. The proposed method makes direct use of the successful ideas of soft margin in support vector machines to improve generalization capability and decision feedback learning in minimum classification error training to enhance model separation in classifier design. SME is illustrated from a perspective of statistical learning theory. By including a margin in formulating the SME objective function, SME is capable of directly minimizing an approximate test risk bound. Frame selection, utterance selection, and discriminative separation are unified into a single objective function that can be optimized using the generalized probabilistic descent algorithm. Tested on the TIDIGITS connected digit recognition task, the proposed SME approach achieves a string accuracy of 99.43%. On the 5k-word Wall Street Journal task, SME obtains relative word error rate reductions of about 10% over our best baseline results in different experimental configurations. We believe this is the first attempt to show the effectiveness of margin-based acoustic modeling for large vocabulary continuous speech recognition in a hidden Markov models framework. Further improvements are expected because the approximate test risk bound minimization principle offers a flexible and rigorous framework to facilitate incorporation of new margin-based optimization criteria into hidden Markov model training.**

***Index Terms*—soft margin estimation, test risk, statistical learning, discriminative training**

I. Introduction

With the prevailing usage of hidden Markov models (HMMs), rapid progress in automatic speech recognition (ASR) has been witnessed in the last two decades. Usually, the HMM parameters are estimated by the traditional maximum likelihood estimation (MLE) method. MLE is known to be optimal for density estimation, but it often does not lead to minimum recognition error that is the goal of ASR. As a remedy, several discriminative training (DT) methods have been proposed in recent years to boost the ASR system accuracy. Typical methods are maximum mutual information estimation (MMIE) [1], [2], [3]; minimum classification error (MCE) [4], [5], [6]; and minimum word/phone error (MWE/MPE) [7]. MMIE training separates different classes by maximizing approximate posterior probabilities. On the other hand, MCE directly minimizes approximate string errors, while MWE/MPE attempts to optimize approximate word and phone error rates. If the acoustic conditions in the testing set match well with those in the training set, these DT algorithms usually achieve very good performance when tested. However, such a good match cannot always be expected for most practical recognition conditions. To avoid the problem of over-fitting on the training set, regularization is achieved by using "I-smoothing" [7] in MMIE and MWE/MPE while MCE exploits a smoothing parameter in a sigmoid function for regularization [8].

According to statistical learning theory [9], a test risk is bounded by the summation of two terms: an empirical risk (i.e., the risk on the training set) and a generalization function. The power to deal with possible mismatches between the training and testing conditions can often be measured by the generalization function. In particular, large margin learning frameworks, such as support vector machines (SVMs) [10], have demonstrated superior generalization abilities over other conventional classifiers. By securing a margin from the decision boundary to the nearest training sample, a correct decision can still be made if the mismatched test sample falls within a tolerance region around the original training samples defined by the margin. The idea of SVMs is explored widely in speech research. Different kinds of kernels are employed in the area of speaker recognition, such as the work in [11], [12]. However, this kind of work cannot be easily incorporated into ASR because it is hard to combine with HMMs. SVMs were also used in the framework of landmark-based speech detection [13]; however this framework is not widely used because it deviates from the HMM paradigm. Some technologies (e.g., [14], [15]) loosely couple SVMs with HMMs by using SVMs instead of Gaussian mixture models (GMMs) as the state observation density of HMMs. These frameworks do not take full advantage of SVMs to get better generalization with a larger margin. A combination of SVMs and HMMs, called

This work was partially supported by the NSF grant, IIS-04-27113, the GALE program of the Defense Advanced Research Projects Agency, Contract No. HR0011-06-C-0022, and the NSF grant DMS-0624841.

Jinyu Li and Chin-Hui Lee are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. 30332 USA (e-mail: {jinyuli, chl}@ece.gatech.edu).

Ming Yuan is with the School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA. 30332 USA (e-mail: myuan@isye.gatech.edu).

HM-SVMs, was explored in [16] with discrete distributions, but it is far from being a solution to the state-of-the-art ASR systems, whose state distributions are usually continuous densities. Moreover, like SVMs, HM-SVMs work on the problem of finding the optimal projection matrix. HM-SVMs differ from SVMs on the point that the observations of HM-SVMs are sequences instead of discrete samples. Therefore, HMMs are used to model the hidden state of the sequences in HM-SVMs. Obviously, HM-SVMs are too simple to solve the ASR problem.

Adopting the concept of enhancing margin separation, large margin estimation (LME) [17], [18] and its variant, large relative margin estimation (LRME) [19], of HMMs have been proposed. In essence, LME and LRME update models only with accurately classified samples. However, it is well known that misclassified samples are also critical for classifier learning. Recently, LRME was modified [20] to consider all of the training samples, especially to move the most incorrectly classified sample toward the direction of correct decision. However, this modification makes the algorithm vulnerable to outliers, and the idea of margin is not very meaningful. In [21], a large margin algorithm for learning GMMs was proposed, but makes some approximations to use GMMs instead of HMMs. More recently, the work of [21] was extended to deal with HMMs in [22], by summing the differences of Mahalanobis distances [23] between the models in the correct and competing string, and comparing the result with a Hamming distance. It is not clear whether it is suitable to directly compare the Hamming distance (the number of different labels of two strings) with the Mahalanobis distance difference, which is the distance of two Gaussian models given an observation.

In this study, soft margin estimation (SME) is proposed as a unified DT framework for discriminative separation, frame selection, and utterance selection. Frame/utterance selection is to select the critical frames/utterances for SME training, instead of using all the training frames/utterances. Because of the incorporation of a soft margin into the optimization objective, SME achieves a better generalization capability and less recognition errors over LME and MCE. We illustrate the SME theory and show that its objective function approximates a bound of the test risk expressed as a sum of an empirical risk and a function of Vapnik & Chervonenkis dimension, or VC dimension, commonly known in statistical learning theory [9]. SME is in contrast to most DT methods that attempt to minimize the empirical risks with additional strategies for generalization. SME is also different from LME because LME only maximizes the separation margin. We show that different choices of separation measures in loss functions lead to various approximate test risks that can be formulated as functions of string, word and phone errors and their combinations. This makes SME flexible and capable of incorporating new losses and margin definitions in a theoretically rigorous manner.

Using 12-state digit models, SME achieves a string accuracy of 99.43%, the best result ever reported on the TIDIGITS database [24] using 32-component mixture Gaussian state observation densities when no further decoding option is used. Even with 1-mixture SME models, the achieved string accuracy is better than that obtained with 32-mixture MLE models, although a single Gaussian model cannot characterize the state distribution well.

The effectiveness of SME was also evaluated on the 5k-word Wall Street Journal (5k-WSJ0) task [25]. Two separation measures are proposed to take advantage of competing strings in lattices obtained from speech recognition. One method is similar to current DT algorithms, defining corresponding separation measures with statistics collected from a lattice using forward backward methods. The other method defines separation using word pairs appearing in a lattice. The performance of the second method (SME_word) is compared with those of MLE and MCE. Initial results on the 5k-WSJ0 task show that SME_word outperforms both MLE and MCE, with around 10% relative word error rate reduction from the MLE baselines. Further performance improvements are expected with flexible combinations of loss and margin function definitions.

II. Empirical Risk and Test Risk Bound

In this section, we show that there is a gap between empirical risk and test risk. The theory of statistical learning explains this gap and gives insight of current state-of-the-art HMM learning algorithms for designing ASR systems.

A. Empirical Risk

The purpose of classification and recognition is usually to minimize classification errors on a representative testing set by constructing a classifier f (modeled by the parameter set Λ) based on a set of training samples $(x_1, y_1),\ldots, (x_N, y_N) \in X * Y$. X is the observation space, Y is the label space, and N is the number of training samples. However we do not know exactly what the property of testing samples is and can only assume that the training and testing samples are independently and identically distributed from some distribution $P(x, y)$. Therefore, we want to minimize the expected classification risk:

$$R(\Lambda) = \int_{X*Y} l(x, y, f_\Lambda(x, y)) dP(x, y).$$

$l(x, y, f_\Lambda(x, y))$ is a loss function. There is no explicit knowledge of the underlying distribution $P(x, y)$. It is convenient to assume that there is a density $p(x, y)$ corresponding to the distribution $P(x, y)$, and replace $\int dP(x, y)$ with $\int p(x, y) dxdy$. Then, $p(x, y)$ can be approximated with the empirical density as:

$$p_{emp}(x, y) = \frac{1}{N}\sum_{i=1}^{N} \delta(x, x_i)\delta(y, y_i),$$

where $\delta(x, x_i)$ is the Kronecker delta function. Finally, the empirical risk is minimized instead of the intractable expected risk:

$$R_{emp}(\Lambda) = \int_{X*Y} l(x, y, f_\Lambda(x, y)) p_{emp}(x, y) dxdy = \frac{1}{N}\sum_{i=1}^{N} l(x_i, y_i, f_\Lambda(x_i, y_i))$$

TABLE I
DISCRIMINATIVE TRAINING TARGET FUNCTION AND LOSS FUNCTION

	Optimization Objective	Loss Function l
MMIE	$\max \frac{1}{N}\sum_{i=1}^{N}\log\frac{P_\Lambda(O_i\|S_i)P(S_i)}{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)}$	$1-\log\frac{P_\Lambda(O_i\|S_i)P(S_i)}{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)}$
MCE	$\min \frac{1}{N}\sum_{i=1}^{N}\frac{1}{1+\exp(-\gamma h_i(O_i,\Lambda)+\theta)}$	$\frac{1}{1+\exp(-\gamma h_i(O_i,\Lambda)+\theta)}$,
MPE	$\max \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)RawPhoneAccuracy(\hat{S}_i)}{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)}$	$1-\frac{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)RawPhoneAccuracy(\hat{S}_i)}{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)}$

Most current DT methods focus on how to minimize this empirical risk. However, as shown above, the empirical risk approximates the expected risk by replacing the underlying density with its corresponding empirical density. Simply minimizing the empirical risk does not necessarily minimize the expected test risk.

In the application of speech recognition, most DT methods directly minimize the risk on the training set, i.e. the empirical risk, which is defined as:

$$R_{emp}(\Lambda)=\frac{1}{N}\sum_{i=1}^{N}\ell(O_i,\Lambda),$$

where $\ell(O_i,\Lambda)$ is a loss function for utterance O_i, and N is the total number of training utterances. $\Lambda=(\pi,a,b)$ is a parameter set denoting the set of initial state probability, state transition probability and observation distribution. Table I lists the optimization objectives and loss functions of MMIE, MCE and MPE with S_i being the correct transcription and $\hat{S}_i$ denoting the possible string sequence for utterance O_i. In MMIE and MPE, $P_\Lambda(O_i|\hat{S}_i)$ (or $P_\Lambda(O_i|S_i)$) and $P(\hat{S}_i)$ (or $P(S_i)$) are acoustic and language model scores, respectively. $RawPhoneAccuracy(\hat{S}_i)$ is the phone accuracy of the string $\hat{S}_i$ comparing with the ground truth S_i. In MCE, h_i is a misclassification measure defined as the difference between a geometrical average of log likelihoods of competing strings and log likelihood of the correct string. γ and θ are parameters for a sigmoid function. With these loss functions, these DT methods all attempt to minimize some empirical risks.

B. Test Risk Bound

The optimal performance on the training set does not guarantee the optimal performance on the testing set. This stems from the statistical learning theory [9], which states that with at least a probability of $1-\delta$ (δ is a small positive number) the risk on the test set (i.e., the test risk) is bounded as follows:

$$R(\Lambda)\le R_{emp}(\Lambda)+\sqrt{\frac{1}{N}\left(VC_{\dim}\left(\log(2N/VC_{\dim})+1\right)-\log\left(\frac{\delta}{4}\right)\right)}. \quad (1)$$

N is the number of training samples. VC_{dim} is the VC dimension that characterizes the complexity of a classifier function group G, and means that at least one set of VC_{dim} (or less) number of samples can be found such that G shatters them. That the function group G shatters samples B means if samples B are divided into two classes, we always have one function from G, which can correctly classify all the samples into those two classes. Eq. (1) shows that the test risk is bounded by the summation of two terms. The first is the empirical risk, and the second is a generalization (regularization) term which is a function of the VC dimension. Although the risk bound is not strictly tight [10], it still gives us insight to explain current technologies in ASR:

- **The use of more data**: In current large scale large vocabulary continuous speech recognition (LVCSR) tasks, thousands of hours of data may be used to get better performance. This is a simple but effective method. When the amount of data is increased, the empirical risk is usually not changed, but the generalization term decreases as the result of increasing N.
- **The use of more parameters**: With more parameters, the training data will be fit better with reduced empirical risk. However, the generalization term increases at the same time as a result of increasing VC_{dim}. This is because with more parameters, the classification function is more complex and has ability to shatter more training points. Hence, by using more parameters, there is a potential danger of over-fitting when the empirical error does not drop while the generalization term keeps increasing.
- **Most DT methods**: DT methods, such as MMIE, MCE, and MWE/MPE in Table I, focus on reducing the empirical risks and do not consider decreasing the generalization term in Eq. (1) from the perspective of statistical learning theory. However, these DT methods have other strategies to deal with the problem of over-training. "I-smoothing", used in MMIE and MWE/ MPE, makes an interpolation between the objective functions of MLE and the discriminative methods. The sigmoid function of MCE can be interpreted as the integral of a Parzen kernel, helping MCE for regularization. Parzen estimation has the attractive property that it converges when the number of training sample grows to infinity. In contrast, margin-based methods reduce the test risk from the viewpoint of statistical learning theory with the help of Eq. (1).

III. SOFT MARGIN ESTIMATION

In this section, soft margin estimation is proposed as a link between statistical learning theory and ASR. We provide a theoretical perspective about SME, showing that SME directly

minimizes an approximate test risk bound. The idea behind the choice of the loss function for SME is then illustrated and the separation functions are defined. DT algorithms, such as MMIE, MCE, and MWE/MPE, can also be cast in the rigorous SME framework by defining corresponding separation functions. Finally, two solutions to SME are provided and the difference with other margin-based methods is discussed.

A. Approximate Test Risk Bound Minimization

If the right hand side of inequality (1) can be directly minimized, it is possible to minimize the test risk. However, as a monotonic increasing function of VC_{dim}, the generalization term can not be directly minimized because of the difficulty to compute VC_{dim}. It can be shown that VC_{dim} is bounded by a decreasing function of the margin [9] (In this paper, margin is used to stand for the width of margin). Hence, VC_{dim} can be reduced by increasing the margin. Now, there are two targets for optimization: one is to minimize the empirical risk, and the other is to maximize the margin. Because the test risk bound of Eq. (1) is not tight, it is not necessary to strictly follow Vapnik's theorem. Instead, the test risk bound can be approximated by combining two optimization targets into a single SME objective function:

$$L^{SME}(\Lambda)=\frac{\lambda}{\rho}+R_{emp}(\Lambda)=\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}\ell(O_i,\Lambda). \qquad (2)$$

ρ is the soft margin, and λ is a coefficient to balance the soft margin maximization and the empirical risk minimization. A smaller λ corresponds to a higher penalty for the empirical risk. The soft margin usage originates from the soft margin SVMs, which deal with non-separable classification problems. For separable cases, margin is defined as the minimum distance between the decision boundary and the samples nearest to it. As shown in Figure 1, the soft margin for non-separable case can be considered as the distance between the decision boundary (solid line) and the class boundary (dotted line). The class boundary is the same definition as for the separable case after removing the tokens near the decision boundary, and treating these tokens differently using slack variable ε_i in Figure 1. The approximate test risk is minimized by minimizing Eq. (2). Again, this approximate test risk is very rough but helpful for classifier design, according to the analysis in Section II.B.

This view distinguishes SME from both ordinary DT methods and LME. Ordinary DT methods only minimize the empirical risk $R_{emp}(\Lambda)$ with additional generalization tactics. LME only reduces the generalization term by minimizing λ/ρ in Eq. (2), and its margin ρ is defined on correctly classified samples.

B. Loss Function Definition

The next issue is to define the loss function $\ell(O_i,\Lambda)$ for Eq. (2). As shown in Figure 1, the essence of margin-based method is to use a margin to secure some generalization in classifier learning. If the mismatch between the training and testing causes a shift less than this margin, a correct decision can still be made. So, a loss happens only when $d(O_i,\Lambda)$ (the separation between the correct and competing string. It will be defined in Section III.C) is less than the value of the soft margin. It should be emphasized that the loss here is not the recognition error. A recognition error happens when $d(O_i,\Lambda)$ is less than 0. Therefore, the loss function can be defined as:

$$\ell(O_i,\Lambda)=(\rho-d(O_i,\Lambda))_+ =\begin{cases}\rho-d(O_i,\Lambda), & \text{if } \rho-d(O_i,\Lambda)>0,\\ 0, & \text{otherwise}\end{cases}$$

with the SME objective function re-written as:

$$L^{SME}(\rho,\Lambda)=\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}(\rho-d(O_i,\Lambda))_+ =\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}(\rho-d(O_i,\Lambda))I(O_i\in U), \qquad (3)$$

where I is an indicator function, and U is the set of utterances that have the separation measures less than the soft margin.

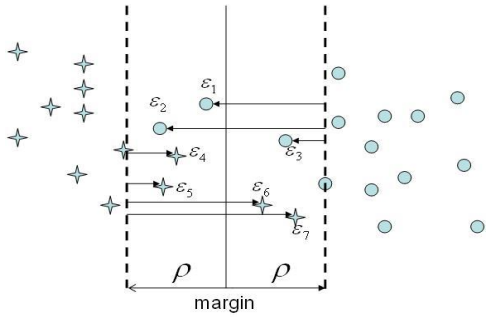

Figure 1. Soft margin estimation. ε_i is the loss of sample *i*, which equals to $\rho-d(O_i,\Lambda)$.

C. Separation Measure Definition

The third step is to define a separation (misclassification) measure, $d(O_i,\Lambda)$, which is a distance between the correct and competing hypotheses. A common choice is to use LLR, or log likelihood ratio, as in MCE [4] and LME [17]:

$$d^{LLR}(O_i,\Lambda)=\log\left[\frac{P_\Lambda(O_i|S_i)}{P_\Lambda(O_i|\hat{S}_i)}\right].$$

If d_i is greater than 0, the classification is correct, otherwise a wrong decision is obtained. $P_\Lambda(O_i|S_i)$ and $P_\Lambda(O_i|\hat{S}_i)$ are the likelihood scores for the target and the most competitive string.

In the following, a more precise model separation measure is defined. For every utterance, we select the frames that have different HMM model labels in the target and competitor string. These frames can provide discriminative information. The model separation measure for a given utterance is defined as the average of those frame LLRs. We use n_i to denote this number of different frames for utterance O_i. Then, a separation of the models is defined as:

$$d^{SME\_utter}(O_i,\Lambda)=\frac{1}{n_i}\sum_j\log\left[\frac{P_\Lambda(O_{ij}|S_i)}{P_\Lambda(O_{ij}|\hat{S}_i)}\right]I(O_{ij}\in F_i), \qquad (4)$$

where F_i is the frame set in which the frames have different labels in the competing strings. O_{ij} is the jth frame for utterance O_i. Only the most competitive string is used in the definition of Eq. (4).

Our separation measure definition is different from LME or MCE, in which the utterance LLR is used. For the usage in SME, the normalized LLR may be more discriminative, because the utterance length and the number of different models in the competing strings affect the overall utterance LLR value. For example, it may not be appropriate that an utterance consisting of five different units in the target and competitive string has greater separation for models inside it than another utterance with only one different unit because the former has a larger LLR value.

By plugging the quantity in Eq. (4) into Eq. (3), the optimization function of SME becomes:

$$L^{SME}(\rho,\Lambda)=\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}\left(\rho-\frac{1}{n_i}\sum_j\log\left[\frac{P_\Lambda(O_{ij}|S_i)}{P_\Lambda(O_{ij}|\hat{S}_i)}\right]I(O_{ij}\in F_i)\right)I(O_i\in U). \quad (5)$$

As shown in Eq. (5), frame selection (by $I(O_{ij}\in F_i)$), utterance selection (by $I(O_i\in U)$), and discriminative separation are unified in a single objective function. This quantity provides a flexible framework for future studies. For example, for frame selection, F_i can be defined as a subset with frames more critical for discriminating HMM models, instead of equally choosing distinct frames in current study.

We can also define separations corresponding to MMIE, MCE, and MPE as shown in Table II. These separations will be studied in future. All these measures can be put back into Eq. (3) for HMM parameter estimation.

TABLE II
SEPARATION MEASURES FOR SME

$d^{SME\_utter}(O_i,\Lambda)$	$\frac{1}{n_i}\sum_j\log\left[\frac{P_\Lambda(O_{ij}\|S_i)}{P_\Lambda(O_{ij}\|\hat{S}_i)}\right]I(O_{ij}\in F_i)$
$d^{SME\_MMIE}(O_i,\Lambda)$	$\log\frac{P_\Lambda(O_i\|S_i)P(S_i)}{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)}$
$d^{SME\_MCE}(O_i,\Lambda)$	$1-\frac{1}{1+\exp(-\gamma h_i(O_i,\Lambda)+\theta)}$
$d^{SME\_MPE}(O_i,\Lambda)$	$\frac{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)RawPhoneAcuracy(\hat{S}_i)}{\sum_{\hat{S}_i}P_\Lambda(O_i\|\hat{S}_i)P(\hat{S}_i)}$

D. Solutions to SME

In this section, two solutions to SME are proposed. One solution is to optimize the soft margin and the HMM parameters jointly. The other is to set the soft margin in advance and then find the optimal HMM parameters. We will show in theory there is little difference between these two methods. And the experiment results in Section V.A will also demonstrate it.

1) *Jointly optimize the soft margin and the HMM parameters:* In this solution, the indicator function $I(O_i\in U)$ in Eq. (3) is approximated with a sigmoid function. Then Eq. (3) becomes:

$$L^{SME}(\rho,\Lambda)=\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}(\rho-d^{SME}(O_i,\Lambda))\frac{1}{1+\exp(-\gamma(\rho-d^{SME}(O_i,\Lambda)))}, \quad (6)$$

where γ is a smoothing parameter for the sigmoid function. Eq. (6) is a smoothing function of the soft margin ρ and the HMM parameters Λ. Therefore, these parameters can be optimized by iteratively using the generalized probabilistic descent (GPD) algorithm on the training set as in [26], with η_t and κ_t as step sizes:

$$\begin{cases}\Lambda_{t+1}=\Lambda_t-\eta_t\nabla L^{SME}(\rho,\Lambda)|_{\Lambda=\Lambda_t}\\ \rho_{t+1}=\rho_t-\kappa_t\nabla L^{SME}(\rho,\Lambda)|_{\rho=\rho_t}\end{cases}.$$

Similar to the solution to soft margin SVMs [10], we need to preset the coefficient λ, which balances the soft margin maximization and the empirical risk minimization.

2) *Presetting the soft margin and optimize the HMM parameters*: Let $(\hat{\rho},\hat{\Lambda})$ be the solution to SME with $\hat{d}_i=d(O_i,\hat{\Lambda})$. Here $d(O_i,\hat{\Lambda})$ can be any separation measure defined in Table II. Then $\hat{\rho}$ minimizes the following quantity:

$$\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}(\rho-\hat{d}_i)_+.$$

Equivalently, $\hat{\rho}$ is a solution to the following:

$$\min\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}u_i \quad \text{subject to} \quad u_i\ge 0,\ u_i\ge\rho-\hat{d}_i.$$

Next, we show that there is a correspondence between λ and $\hat{\rho}$.

There exist nonnegative constants, α_i and β_i, such that $\hat{\rho}$ also minimizes

$$L^{\rho}=\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}u_i-\sum_{i=1}^{N}\alpha_i u_i-\sum_{i=1}^{N}\beta_i(u_i-\rho+\hat{d}_i). \quad (7)$$

Eq. (7) is the Lagrange form [27] for SME, and is known to be equivalent to the original optimization problem. The first order condition implies that

$$\frac{\partial L^{\rho}}{\partial u_i}=\frac{1}{N}-\alpha_i-\beta_i=0, \quad (8)$$

$$\frac{\partial L^{\rho}}{\partial\rho}=-\frac{\lambda}{\rho^2}+\sum_{i=1}^{N}\beta_i=0. \quad (9)$$

Incorporating the derivatives in Eqs. (8) and (9) into Eq. (7), we have ($\hat{\beta}$, $\hat{\rho}$) as the solution to

$$\min\sum_{i=1}^{N}-\beta_i\hat{d}_i+2\sqrt{\lambda\sum_{i=1}^{N}\beta_i} \quad \text{subject to} \quad 0\le\beta_i\le\frac{1}{N},\ \sum_{i=1}^{N}\beta_i=\frac{\lambda}{\rho^2}.$$

Now, let $\tilde{\beta}$ be the solution to

$$\min \sum_{i=1}^{N} -\beta_i \hat{d}_i + 2\sqrt{\lambda \sum_{i=1}^{N} \beta_i} \quad .$$
$$\text{subject to} 0 \le \beta_i \le \frac{1}{N}$$

Then, the solution to the original problem is

$$\hat{\beta} = \tilde{\beta}, \qquad \hat{\rho} = \sqrt{\frac{\lambda}{\sum_{i=1}^{N} \tilde{\beta}_i}} \; . \tag{10}$$

On the other hand, $\hat{\Lambda}$ minimizes

$$L^{\Lambda} = \sum_{i=1}^{N} \left(\hat{\rho} - d(O_i, \Lambda)\right)_+ \; . \tag{11}$$

From Eq. (10), we see a direct mapping relationship between λ and $\hat{\rho}$. For a fixed λ, there is one corresponding $\hat{\rho}$. Instead of choosing a fixed λ and trying to get the solution of $\left(\hat{\rho}, \hat{\Lambda}\right)$ as in the first solution, we can directly choose a $\hat{\rho}$ in advance and get $\hat{\Lambda}$ by minimizing Eq. (11) because of the mapping relationship between λ and $\hat{\rho}$. There is no explicit knowledge what λ should be, so it is not necessary to start from λ and get the exact corresponding solution of $\hat{\rho}$. In contrast, we will show in the section of experiments that it is easy to draw some knowledge of the range of $\hat{\rho}$. Setting $\hat{\rho}$ in advance is a simple way to solve the SME problem.

Because of a fixed $\hat{\rho}$, only the samples with separation smaller than the margin need to be considered. Assuming that there are a total of N_C utterances satisfying this condition, we can minimize the following with the constraint $d(O_i, \Lambda) < \hat{\rho}$:

$$L^{sub}(\Lambda) = \sum_{i=1}^{N_C} \left(\hat{\rho} - d(O_i, \Lambda)\right) \; . \tag{12}$$

Now, this problem can be solved by the GPD algorithm by iteratively working on the training set, with η_t as a step size: $\Lambda_{t+1} = \Lambda_t - \eta_t \nabla L^{sub}(\Lambda)|_{\Lambda = \Lambda_t}$.

E. Margin-Based Methods Comparison

In this section, SME is compared with two margin-based method groups. One group is LME [17], [18], [28], and the other is large margin GMM (LM-GMM) [21] and large margin HMM (LM-HMM) [22]. LM-HMM and LM-GMM are very similar, except that LM-HMM measures model distance in a whole utterance while LM-GMM measures in a segment. The difference of these margin-based methods is listed in Table III and is discussed in the following.

- **Training sample usage**: Both LM-GMM/LM-SVM and SME use all the training samples, while LME only uses correctly classified samples. The misclassified samples are important for classifier learning because they carry the information to discriminate models. Except for LME, DT methods usually use all the training samples (e.g. [1]-[7]).
- **Separation measure**: It is crucial to define a good separation measure because it directly relates to margin. LME uses utterance based LLR as a measure; while in SME it is carefully represented by a normalized LLR measure over only the set of different frames. With such normalization, the utterance separation values can be more closely compared with a fixed margin than an un-normalized LLR without being affected by different numbers of distinct units and length of the utterances. LM-GMM and LM-HMM use Mahalanobis distance [23], which makes it hard to be directly used in the context of mixture models. In [21] and [22], approximation to the mixture component with the highest posterior probability under GMM is applied.
- **Segmental training**: Speech is segment based. Both SME and LME use HMMs, while LM-GMM uses frame averaged GMM to approximate segmental training. As an improvement, LM-HMM directly works on the whole utterance. It sums the difference of the Mahalanobis distances between the models in the correct and competing string, and compares with a Hamming distance. That Hamming distance is the number of mismatched labels of recognized string. Although similar distance (raw phone accuracy) has been used in MPE [7] for weighting the contribution different recognized strings, it is not clear whether Hamming distance is suitable to be directly used to compare with the Mahalanobis distance because these two distances are very different types of measures (one is for string labels and the other is for Gaussian models).
- **Target function**: SME maximizes the soft margin penalized with the empirical risk as in Eq. (2). This objective directly relates to the test risk bound shown in Eq. (1). LME only maximizes its margin, assuming the empirical risk is 0. The idea of LME is to define the minimum positive separation distance as a margin and then maximize it. Because of this, the technology dealing with misclassified samples by making usage of a soft margin or slack variable can not be easily incorporated in LME. LM-GMM/LM-HMM minimizes the summation of all the traces of Gaussian models, penalized with a Mahalanobis distance based misclassification measure.

TABLE III
COMPARISON OF MARGIN BASED METHODS

	LME	LM-GMM [21] LM-HMM [22]	SME
Training Samples	correctly classified samples	all samples	all samples
Separation Measure	utterance LLR	Mahalanobis distance	LLR with frame selection
Segmental Modeling	HMM	frame averaged GMM [21] HMM [22]	HMM
Target Function	margin maximization	penalized trace minimization	penalized margin maximization
Convex Problem	No [17], [18] Yes [28]	Yes	No

- **Convex problem**: LME has several different solutions. In [17], [18], the target function is non-convex. By using a series of transformations and constraints [28], LME can have a convex target function. Also, LM-GMM and LM-HMM formularize their target function as a convex one. The convex function has the nice property that its local minimum is global minimum. This will make the

parameter optimization much easier. To get a convex target function, it needs to approximate the GMM with a single mixture component of the GMM. In contrast, the target function of SME is not convex. Therefore, SME is subject to local minima like most other DT methods. In future, we will investigate whether SME can also get a convex target function with the cost of approximation and some transformations.

IV. SME FOR LVCSR

The key issue for using SME in LVCSR is to define appropriate model separation measures. One method is to directly use $d_i^{SME\_utter}(O_i,\Lambda)$ in Table II, and solve for HMM parameters by minimizing the quantity in Eq. (3). However, most successful DT methods on LVCSR use lattices to get a rich set of competing candidate information. The advantage can also be explained by the test risk bound in Eq. (1) since lattices provide more confusion patterns (i.e., more data). As discussed in Section II. B, this will result in a reduced generalization term, which makes the test risk bound tighter. In the following, two solutions are provided for lattice-based separation measure definition for LVCSR.

A. Utterance Level Separation Measure

The first solution is similar to lattice-based MMIE [3], [29], MCE [30], and MPE [7]. We then define distances, $d^{SME\_MMIE}(O_i,\Lambda)$, $d^{SME\_MCE}(O_i,\Lambda)$, and $d^{SME\_MPE}(O_i,\Lambda)$, as shown in Table II. We can now take advantage of optimization algorithms adopted in lattice-based DT methods to obtain statistics at the utterance level and then use extended Baum-Welch algorithms to optimize parameters. However, due to its focus on utterance level competition, it is possible to lose the advantage of the frame-level discrimination power in the SME separation measures as discussed in the previous section.

B. Word Level Separation Measure

SME separations can also be defined at the word segment level. The first step is to align the utterance with the correct transcription and get the timing information for every word. The second step is to find competing words for every word in the lattice. This is done by examining the lattice to get words falling into the time segment of current correctly transcribed words. A frame overlapping threshold is set not to consider words with too few overlapping frames as competing words. For example, for the lattice in Figure 2, the competing words are listed in Table IV. For the *p*th overlapping word pair, we denote the number of overlapped frames as n_{op}, the *j*th overlapping frame as O_{oj}, set of overlapping frames as F_{op}, and the target and competing words *as* W_{target} and W_{comp}. A word level separation can be defined as:

$$d_{op}^{SME\_word}(O_i,\Lambda)=\frac{1}{n_{op}}\sum_j \log\left[\frac{P_\Lambda\left(O_{oj}\middle|W_{\text{target}}\right)}{P_\Lambda\left(O_{oj}\middle|W_{\text{comp}}\right)}\right] I\left(O_{oj}\in F_{op}\right), \qquad (13)$$

where $P_\Lambda\left(O_{oj}\middle|W_{\text{target}}\right)$ and $P_\Lambda\left(O_{oj}\middle|W_{\text{comp}}\right)$ are the likelihood scores for W_{target} and W_{comp}.

For any word pair W_{target} and W_{comp}, we compute Eq. (13), and plug all of them into the following formula:

$$L^{SME}(\rho,\Lambda)=\frac{\lambda}{\rho}+\frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{np_i}\sum_{p=1}^{np_i}\left(\rho-d_{op}^{SME\_word}(O_i,\Lambda)\right)_+\right], \qquad (14)$$

where np_i denotes the number of overlapping word pairs in utterance O_i.

It should be noted that the indicator functions for frame selection in Eqs. (4) and (13) are discontinuous. Therefore, it is possible that a change in Λ to improve separation may lead to a different model label sequence for the strings, which may in fact lead to a worsening of separation. In Eq. (6), the indicator function for utterance selection is approximated by a sigmoid function. This may be applied for the indicator function of frame selection in future study to ensure continuousness.

We found the word level separation ($d_o^{SME\_word}(O_i,\Lambda)$ with word pairs in lattices) to be better than the utterance level measure ($d_i^{SME\_utter}(O_i,\Lambda)$ with only the correct and most competitive strings), because it uses more confusion patterns. For usage in SME, $d_o^{SME\_word}(O_i,\Lambda)$ may also have an advantage over other separation measures defined above, which have only one value for each utterance. This is because in SME we will plug this separation value into Eq. (3), and the utterances with values greater than the value of the margin will not contribute to parameter optimization. However in some cases, there may be some word pairs in lattices that still have distances less than the value of the margin. The word level separation measure $d_o^{SME\_word}(O_i,\Lambda)$ makes use of those word pairs to get more confusion patterns.

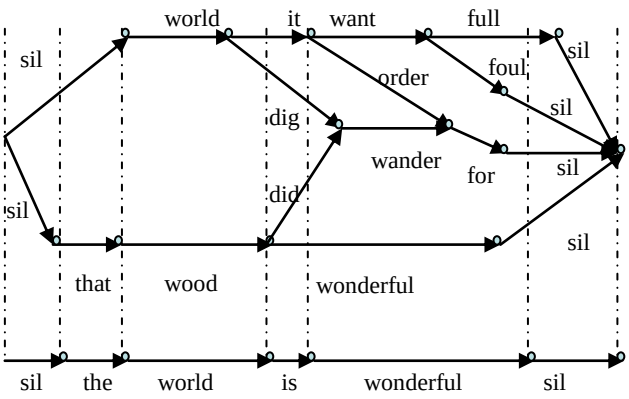

Figure 2. A lattice example: the top lattice is obtained in decoding, and the bottom is the corresponding utterance transcription. "sil" stands for silence.

TABLE IV
CORRECT AND COMPETING WORDS FOR LATTICE EXAMPLE

Correct Word	Competing Words
the	that
world	wood, it, dig
is	it, dig, did, wonderful
wonderful	want, full, foul, order, dig, did, wander, for

V. SPEECH RECOGNITION EXPERIMENTS

The proposed SME framework was evaluated on two

different tasks: the TIDIGITS connected digit and 5k-WSJ0 LVCSR tasks. Using 12-state digit models, SME achieves a string accuracy of 99.43% on the TIDIGITS database. In 5k-WSJ0 LVCSR task, SME gets around 10% relative word error rate reductions from the MLE baselines.

A. *TIDIGITS*

For the TIDIGITS database, the same experimental configuration was used as that in [18]. There are 8623 digit strings in the training set and 8700 digit strings for testing. The hidden Markov model toolkit (HTK) [31] was first used to build the baseline MLE HMMs. There were 11 whole-digit HMMs: one for each of the 10 English digits, plus the word "oh". Each HMM has 12 states and each state observation density is characterized by a mixture Gaussian density. GMM Models with 1, 2, 4, 8, 16, and 32 mixture components were trained. The input features were 12MFCCs + energy, and their first and second order time derivatives. MCE models were also trained for comparison. *N*-best incorrect strings were used for training. The performance of this choice was better than the implementation with the top incorrect string. Different smoothing parameters were tried and the results were with the best one. SME models were initiated with the MLE models. This is in clear contrast with the LME models [17], [18], [28], which are typically built upon the well-performed MCE models. Digit decoding was based on unknown length without imposing any language model or insertion penalty.

TABLE V
MARGIN VALUE ASSIGNMENT

1-mix	2-mix	4-mix	8-mix	16-mix	32-mix
5	6	7.5	8.5	9	11

$d^{SME\_utter}(O_i, \Lambda)$ was used as the separation measure, which means that only the most competitive string was used in SME training. Various soft margin values were set corresponding to different model complexities as shown in Table V. These soft margin values were empirically chosen as the mode of all the separation distances obtained from the MLE model. For example, in Figure 4, the mode of the separation distance of the 1-mixture MLE model is about 5. Therefore, the soft margin value for the 1-mixture SME model was set as 5. Slightly changing values in Table V only made very little difference on final results. While this setting produced satisfactory results, we believe it is too heuristic and suboptimal, and will investigate in future work whether there is any plausible theory underlies it.

Figure 3 shows string accuracy improvement of SME in the training set for different SME models after 200 iterations. Although the initial string accuracies (got from MLE models) were very different, all SME models ended up with nearly the same accuracies of 99.99%. As discussed in Section II, the test risk is bounded by the summation of the empirical risk and the generalization term, which is related with margin. The training errors are nearly the same for all of these different mixture models, and the margin plays significant role in the test risk bound, resulting in different test risks that are listed in Table VI, and to be discussed later.

Figures 4 and 5 compare histograms of the measure defined in Eq. (4) with the normalized LLR for the case of 1-mixture GMM before and after SME. Usually, the larger the separation value, the better the models are. We observe in Figure 5 a very sharp edge around a value of 5, which is the soft margin value for the 1-mixture model update shown in the leftmost column of Table V. It is clear that when SME finishes parameter update, most samples which have separation values less than the specified margin move to the right side of histogram, resulting in separation values greater than the margin value. This demonstrates the effectiveness of the SME algorithms. We can also see the effect in Figure 6, the histogram separation for the 32-mixture case after SME update. The sharp edge now is around 11, the margin shown in the rightmost column in Table V. With a greater margin, the 32-mixture model can attain a string accuracy of 99.43% in the testting set while the 1-mixture model can only get 98.76%, although both models have nearly the same string accuracy in the training set. This observation is greatly in consistent with the test risk bound inequality of Eq. (1).

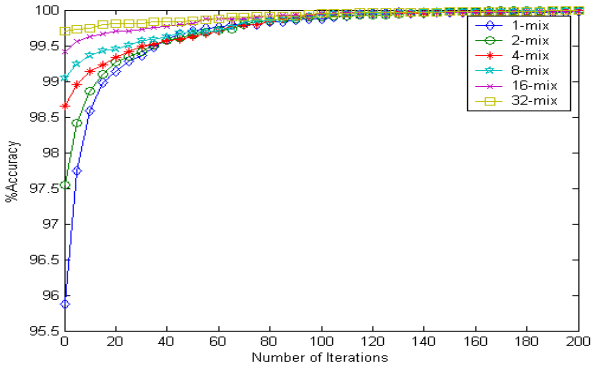

Figure 3. String accuracy of SME for different models in TIDIGITS training.

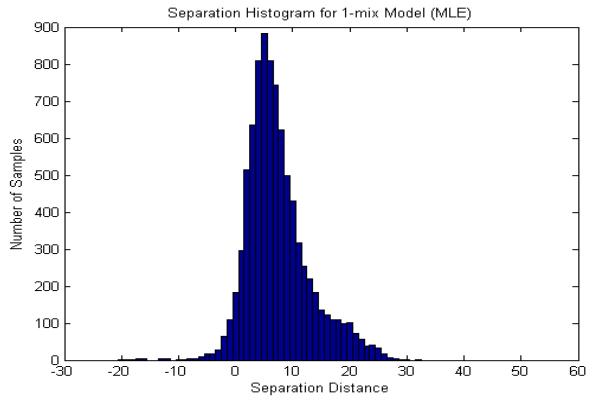

Figure 4. The histogram of separation distance of 1-mixture MLE model in the TIDIGITS training set.

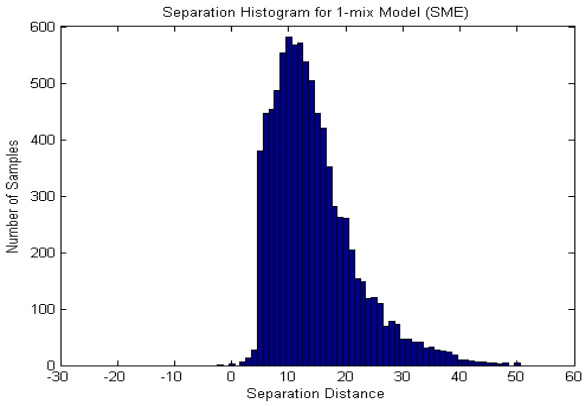

Figure 5. The histogram of separation distance of 1-mixture SME model in the TIDIGITS training set.

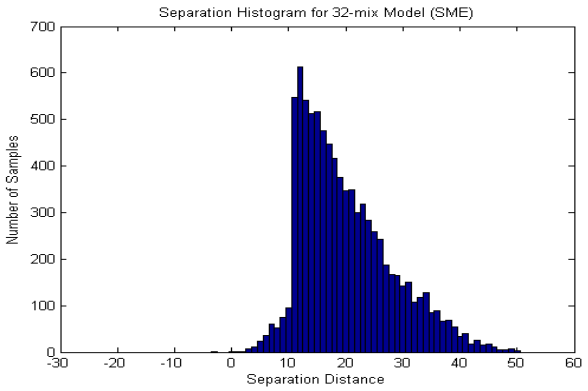

Figure 6. The histogram of separation distance of 32-mixture SME model in the TIDIGITS training set.

TABLE VI
TESTING SET STRING ACCURACY COMPARISON WITH DIFFERENT METHODS. ACCURACIES MARKED WITH AN ASTERISK ARE SIGNIFICANTLY DIFFERENT FROM THE ACCURACY OF THE SME MODEL (P<0.025, PAIRED Z-TEST, 8700 D.O.F. [32]).

	MLE	MCE	LME [18]	SME	SME_joint
1-mix	95.20%*	96.94%*	96.23%*	98.76%	98.74%
2-mix	96.90%*	97.40%*	98.30%*	98.95%	98.92%
4-mix	97.80%*	98.24%*	98.76%*	99.20%	99.11%
8-mix	98.03%*	98.66%*	99.13%	99.29%	99.26%
16-mix	98.36%*	98.87%*	99.18%	99.30%	99.32%
32-mix	98.51%*	98.98%*	99.34%	99.43%	99.40%

Table VI compares different training methods with various numbers of mixture components. Only string accuracies are listed in Table VI. At this high level of performance in TIDIGITS, the string accuracy is a strong indicator of model effectiveness. For the task of string recognition, the interest is usually in whether the whole string is correct. Therefore, string accuracy is more meaningful than the word accuracy in TIDIGITS. Two different solutions of SME are compared in Table VI. The column labeled **SME** presets the soft margin with values defined in Table V. The column labeled **SME_joint** solves SME by optimizing the soft margin and HMM parameters jointly. For the purposed of comparison, the final margin values got by **SME_joint** are listed in Table VII. These values are similar to those margin values preset in Table V. There are only very small differences between the performance of **SME** and **SME_joint** in Table VI. This again demonstrates our opinion in Section III.D that the two proposed solutions are nearly equivalent because of the mapping relationship between λ and $\hat{\rho}$. Because there is some knowledge about the range of $\hat{\rho}$ as in Figure 4 but no explicit knowledge of λ, we prefer setting $\hat{\rho}$ in advance as a simple way to solve SME. In the following sections, unless stated, SME uses the solution that presets the soft margin value.

TABLE VII
MARGIN VALUES GOT BY JOINT OPTIMIZATION

1-mix	2-mix	4-mix	8-mix	16-mix	32-mix
5.2	5.9	7.1	7.4	9.6	10.6

In [4], MCE reduced string error rate from 1.4% (MLE) to 0.95%, using a 10-state 64-mixture whole word models. The MCE performance of our 12-state 32-mixture whole word models is similar to the results in [4], reducing string error rate from 1.49% (MLE) to 1.02%. Clearly SME outperforms MLE and MCE significantly, and is consistently better than LME. For 1-mixture SME models, the string accuracy is 98.76%, which is better than that of the 32-mixture MLE models. The goal of our design is to separate the models as far as possible, instead of modeling the observation distributions. With SME, even 1-mixture models can achieve satisfactory model separation. The excellent SME performance is attributed to the well defined model separation measure and good objective function for generalization.

We believe the string accuracy of 99.43% listed in the bottom row of Table VI represents the best result ever reported on the TIDIGITS task with similar configuration. In [33], 99.45% string accuracy was reported with 32-mixture models by using a grammar and word insertion penalty. In [28], 99.47% string accuracy was obtained with the constraint that the maximum string length is 7 [34]. These decoding constraints typically improve string accuracies. However, they were not used in our experiments.

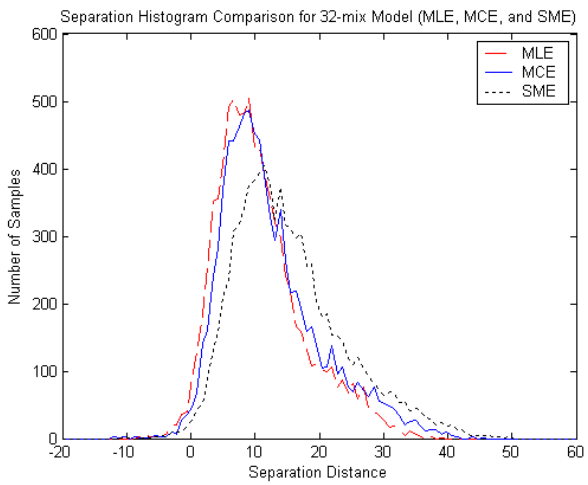

Figure 7. The histogram of separation distance of 32-mix model of MLE, MCE, and SME in the TIDIGITS testing set. The short dashed curve, line curve and dot curve correspond to MLE, MCE, and SME models.

To compare the generalization capability of SME with MLE and MCE, we plot the histograms of the separation measure for the testing utterances in Figure 7 for the 32-mixture MLE, MCE and SME models. As indicated in the rightmost curve, SME achieves a significantly better separation than both MLE and MCE in the testing set, due to direct model separation maximization and better generalization.

B. 5k WSJ0

The 5k-WSJ0 task was used to evaluate the effectiveness of SME on LVCSR. The training material is the SI-84 set, with 7077 utterances from 84 speakers. Testing is performed on the Nov92 evaluation set, with 330 utterances from 8 speakers. Baseline HMMs were within-word triphone models trained with MLE using HTK. There were a total of 2329 shared states obtained with a decision tree, and each state observation density was modeled by an 8-mixture GMM. The input features were 12MFCCs + energy, and their first and second order time derivatives. The bigram and trigram language models (LMs) within the 5k-WSJ0 vocabulary were used for decoding. The baseline WERs were 8.41% with bigram LM and 6.13% with trigram LM, respectively. Both the bigram and the trigram LMs were provided by the original WSJ0 corpus. We note that other studies have reported better results than our baseline systems by using different configurations (ex. in [30]). In this paper, we do not have access to those baseline configurations, so we only attempt to improve over our best available setups. Our HTK-trained baselines are comparable with the HTK-trained within-word triphone results reported in [35] and recent results in [36]. In [37], the WER of 7.87% was reported with cross-word triphone models. Our baseline is also comparable with this result, considering the different within-word and cross-word settings. The proposed SME algorithm is expected to improve over better baseline systems as well.

The bigram LM was used to obtain seed lattices for all of the training utterances. These lattices were generated only once. At each iteration, the recently updated HMMs were incorporated to generate new lattices by using seed lattices as decoding word graphs. Following this, SME was used to update HMM parameters. The method, denoted by **SME_word**, is based on the word level separation measure, $d_o^{SME\_word}(O_i, \Lambda)$, defined in Eq. (11). The soft margin value was set to 5.

MCE model was trained with the similar implementation as [30]. The bigram LM was used to generate lattices and unigram was used to rescore them. The correct path was removed from the decoded word graph, and the smoothing constant was set to 0.04 as in [30]. However, since the relative WER rate of this MCE realization is worse than that reported in [30], there may be some implementation issues we need to investigate.

In Table VIII, the WERs obtained with MLE, MCE and the SME method are compared. The SME method achieved lower WERs than those obtained with the MLE and MCE models. **SME_word** decreased WERs significantly from MLE, with relative WER reductions of 12% for bigram LM and 9% for trigram LM, respectively. These relative WER rates are comparable to that reported in [30]. Therefore, we believe SME can also work on LVCSR as well as other DT methods.

TABLE VIII
PERFORMANCE COMPARISON ON THE 5K-WSJ0 TASK

WER	Bigram	Trigram
MLE	8.41%	6.13%
MCE	7.85%	5.83%
SME_word	7.38%	5.60%

It should be noted that the current implementations of MCE and SME are different. Therefore, there is no safe conclusion whether the formalization of SME is really better than MCE. For the purposed of fair comparison, it is desirable to share most implementations for MCE and SME, differing only with their distinguished algorithm parts. In [38], we formalized SME in string level and shared the most realizations with MCE, and the difference was only on margin-based utterance and frame selection for SME. The results clearly showed SME outperformed MCE with the help of margin. That work is out of the scope of the current paper, please refer [38] for detail.

VI. CONCLUSION AND DISCUSSION

We have proposed a novel discriminative training method, called SME, to achieve both high accuracy and good model generalization. This proposed method utilizes the successful ideas of soft margin in SVMs to improve generalization capability. It directly maximizes the separation of competing models to enhance the testing samples to approach a correct decision if the deviation from training models is within a safe margin. Frame and utterance selections are integrated into a unified framework to select the training utterances and frames critical for discriminating competing models. From the view of statistical learning theory, we show that SME can minimize the approximate risk bound on the test set. The choice of various loss functions is illustrated and different kinds of separation measures are defined under a unified SME framework.

Tested on the TIDIGITS database, even 1-mixture SME models can better separate different words and produce better string accuracy than 32-mixture MLE models. The performance of SME is consistently better than that of LME, and significantly better than those of MLE and MCE. The experiment coincides with the inequality of the test risk bound, showing that even though all the models have the same training errors, the test string accuracies differ because of different margin values associated with various models. SME was also applied to LVCSR by defining separation measures at the word levels. Tested on the 5k-WSJ0 task, SME achieves about 10% relative WER reductions over our best MLE baselines.

This paper represents an initial study; we are now working on a number of related research issues to further complete the theory of SME. The first is to expand the framework of SME. Different options can be integrated into current framework. For example, the current frame selection procedure gives equal weight to all the different frames in the correct and competing strings. Some frames in an utterance may be more critical to measure the separation of models. A strategy to select these

critical frames will be investigated. More elaborate definitions of margin functions will also be explored to tightly couple them with the definition of the empirical risks.

The second research item is to design a better solution to Eq. (3). Two solutions to SME are proposed in this study. One solution is to obtain HMM parameters by presetting the soft margin, and the other is to optimize the soft margin and HMMs together. Because there is a correspondence between the soft margin and the balance coefficient λ, these two solutions are nearly equivalent. The solution of jointly optimizing soft margin and HMM parameters needs to work under a fixed λ. How to select a satisfactory λ is still an open problem in machine leaning. Determining how to obtain the soft margin value in advance for the presetting margin solution is another important problem, although the histogram such as Figure 4 gives good indication. We will explore what the true margin should be and its relationship with the HMM parameters. We will also investigate the theory for better selection of λ. Better performance is expected with more precise selection of the margin value or the balance coefficient λ.

The third item is to implement a better optimization method. As discussed in Section III.E, with some approximation, LME and LM-GMM/LM-HMM convert the original problem into a convex optimization problem. The sacrifice of precision gives a nice convex target function. We will explore this tradeoff, and see whether SME can also be cast into a convex problem.

Finally, we will continue to work on LVCSR. We believe the current WSJ0 performance is far from optimal. We will study the usage of LM in SME training. Currently, LM is only used to generate lattices for utterances and not used for SME parameter update. The relationship between SME and LM will be evaluated in future studies. Another research direction is to take advantage of the other successful DT algorithms by using their corresponding separation measures defined in Table II.

ACKNOWLEDGMENT

We would like to thank Dr. Hui Jiang of York University for valuable discussions on LME. We are grateful to Dr. Sabato Marco Siniscalchi for the lattice phone alignment tool. We thank Zhijie Yan for helping to run MCE experiments on the WSJ0 task. We appreciate Jeremy Reed and Brett Matthews for improving the English presentation of the paper. We also thank the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] L. R. Bahl, P. F. Brown, P.V. de Souza and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," *Proc. ICASSP*, vol. 1, pp. 49-52, 1986.

[2] Y. Normandin, "Maximum mutual information estimation of hidden Markov models," In *Automatic Speech and Speaker Recognition*, C. -H. Lee, F. K. Soong and K. K. Paliwal, Eds. Kluwer Academic Publishers, 1996.

[3] V. Valtchev, J. Odell, P. C. Woodland and S. Young, "MMIE training of large vocabulary recognition systems," *Speech Communication*, vol. 22, no. 4, pp. 303-314, 1997.

[4] B. -H. Juang, W. Chou and C. -H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Proc.*, vol. 5, no. 3, pp. 257-265, 1997.

[5] R. Schlueter, W. Macherey, B. Muller, and H. Ney, "Comparison of discriminative training criteria and optimization methods for speech recognition," *Speech Communication*, vol. 34, no. 3, pp. 287–310, 2001.

[6] E. McDermott and T. J. Hazen and J. L. Roux, and A. Nakamura and S. Katagiri, "Discriminative training for large vocabulary speech," *IEEE Trans. On Audio, Speech, and Language Proc.*, vol. 15, no. 1, pp. 203-223, 2007.

[7] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," *Proc. ICASSP*, pp. I105-I108, 2002.

[8] E. McDermott and S. Katagiri, "A derivation of Minimum Classification Error from the theoretical classification risk using Parzen estimation," *Computer Speech and Language*, vol. 18, pp. 107-122, 2004.

[9] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[10] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

[11] W. Campbell, "Generalized linear discriminant sequence kernels for speaker recognition," *Proc. ICASSP*, pp. 161-164, 2002.

[12] J. Louradour, K. Daoudi and F. Bach, "SVM speaker verification using an incomplete cholesky decomposition sequence kernel," *Proc. IEEE Odyssey*, 2006.

[13] M. H. Johnson, J. Baker, S. Borys, K. Chen, E. Coogan, S. Greenberg, A. Juneja, K. Kirchhoff, K. Livescu, S. Mohan, J. Muller, K. Sonmez and T. Wang, "Landmark-based Speech Recognition: Report of the 2004 Johns Hopkins Summer Workshop", *Proc. ICASSP*, 2005.

[14] A. Ganapathisraju, J. Hamaker and J. Picone, "Hybrid SVM/HMM architecture for speech recognition," *Proc. Interspeech*, pp. 504–507, 2000.

[15] J. Stadermann and G. Rigoll, "A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition," *Proc. Interspeech*, pp. 661–664, 2004.

[16] Y. Altun, I. Tsochantaridis and T. Hofmann, "Hidden Markov support vector machines," *Proc. ICML*, 2003.

[17] X. Li, H. Jiang and C. Liu, "Large margin HMMs for speech recognition," *Proc. ICASSP*, pp. V513-V516, 2005.

[18] H. Jiang, X. Li, and C. Liu, "Large margin hidden Markov models for speech recognition," *IEEE Trans. On Audio, Speech, and Language Proc.*, vol. 14, no. 5, pp. 1584-1595, 2006.

[19] C. Liu, H. Jiang and X. Li, "Discriminative training of CDHMMS for maximum relative separation margin," *Proc. ICASSP*, pp. I101-I104, 2005.

[20] C. Liu, H. Jiang and L. Rigazio, "Recent improvement on maximum relative margin estimation of HMMs for speech recognition," *Proc. ICASSP*, pp. I269-I272, 2006.

[21] F. Sha and L. K. Saul, "Large margin Gaussian mixture modeling for phonetic classification and recognition," *Proc. ICASSP*, pp. I265-I268, 2006.

[22] F. Sha and L. K. Saul, "Large margin hidden Markov models for automatic speech recognition," in Advances in Neural Information Processing Systems 19, B. Sch¨olkopf, J.C. Platt, and T. Hofmann, Eds., MIT Press, 2007.

[23] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, Inc., New York, 2001.

[24] R. G. Leonard, "A database for speaker-independent digit recognition," *Proc. ICASSP*, 1984.

[25] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," *Proceedings of the workshop on Speech and Natural Language*, 1992.

[26] S. Katagiri, B. -H. Juang and C.-H. Lee , "Pattern recognition using a family of design algorithms based upon the generalized probabilistic descent method," *Proc. IEEE*, vol. 86, no. 11, pp. 2345-2373, 1998.

[27] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2000.

[28] X. Li and H. Jiang, "Solving large margin estimation of HMMs via semidefinite programming," *Proc. Interspeech*, pp. 2414-2417, 2006.

[29] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 25–47, 2002.

[30] W. Macherey, L. Haferkamp, R. Schlüter and H. Ney "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," *Proc. Interspeech*, pp. 2133-2136, 2005.
[31] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. C. Woodland, *The HTK Book (for HTK Version 3.2)*, Cambridge University, 2002.
[32] E. L. Lehmann, *Testing Statistical Hypothesis (2nd edition)*, Wiley, 1986.
[33] D. Yu, L. Deng, X. He and A. Acero, "Use of incrementally regulated discriminative margins in MCE training for speech recognition," *Proc. Interspeech*, pp. 2418-2421, 2006.
[34] H. Jiang, York University, Canada, private communication, 2006.
[35] P. C. Woodland, J. Odell, V. Valtchev and S. Young, "Large vocabulary continuous speech recognition using HTK," *Proc. ICASSP*, pp. II125-II128, 1994.
[36] Q. Fu, A. M. Daniel, B. -H. Juang, J. L. Zhou and F. K. Soong, "Generalization of the minimum classification error (MCE) training based on maximizing generalized posterior probability (GPP)," *Proc. Interspeech*, pp. 681-684, 2006.
[37] B. Mak, T. -C. Lai, R. Hsiao, "Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers," *Proc. ICASSP*, pp. I222-I232, 2006.
[38] J. Li, Z. Yan, C. -H. Lee, and R. -H. Wang, "A study on soft margin estimation for LVCSR," submitted to *IEEE Automatic Speech Recognition and Understanding Workshop*, 2007.