Large Margin Hidden Markov Models for Speech Recognition

Hui Jiang, *Member, IEEE*, Xinwei Li, and Chaojun Liu, *Member, IEEE*

***Abstract*—In this paper, motivated by large margin classifiers in machine learning, we propose a novel method to estimate continuous-density hidden Markov model (CDHMM) for speech recognition according to the principle of maximizing the minimum multiclass separation margin. The approach is named large margin HMM. First, we show this type of large margin HMM estimation problem can be formulated as a constrained minimax optimization problem. Second, we propose to solve this constrained minimax optimization problem by using a penalized gradient descent algorithm, where the original objective function, i.e., minimum margin, is approximated by a differentiable function and the constraints are cast as penalty terms in the objective function. The new training method is evaluated in the speaker-independent isolated E-set recognition and the TIDIGITS connected digit string recognition tasks. Experimental results clearly show that the large margin HMMs consistently outperform the conventional HMM training methods. It has been consistently observed that the large margin training method yields significant recognition error rate reduction even on top of some popular discriminative training methods.**

***Index Terms*—Continuous-density hidden Markov models (CDHMMs), gradient descent search, large margin classifiers, minimax optimization, support vector machine.**

I. Introduction

THE most successful modeling approach to automatic speech recognition (ASR) is to use a set of hidden Markov models (HMMs) as the acoustic models for subword or whole-word speech units and to use the statistical N-gram model as language model for words and/or word classes in sentences. All the model parameters, including HMMs and N-gram models, are estimated from a large amount of training data according to certain criterion. It has been shown that success of this kind of data-driven modeling approach highly depends on the goodness of estimated models. As for HMM-based acoustic models, the dominant estimation method is the Baum–Welch algorithm which is based on the maximum likelihood (ML) criterion. As an alternative to the ML estimation, discriminative training (DT) has also been extensively studied for HMMs in ASR. The DT methods aim to minimize or reduce classification errors in training data as model estimation criterion. Over the past decade, it has been experimentally found that the discriminative training methods can improve the ASR performance over the standard ML method in many tasks, such as in [8], [17], [18], [26], [28], [31], [45] and many others. Generally speaking, the maximum mutual information (MMI) [3], [31], [45] and minimum classification error (MCE) [18], [19] formulation have been regarded as the most successful DT methods in ASR. In the MMI formulation, estimation criterion is to maximize the mutual information between training data and their corresponding models (or class labels). A growth-transformation based optimization method, *a.k.a.* extended Baum–Welch (EBW) algorithm, is used to optimize model parameters to achieve the maximum mutual information in order to establish the possibly tightest relation (in a probabilistic sense) between data and their corresponding models. More recently, some variants, such as minimum phone error (MPE) in [32], have also been proposed for ASR under the MMI framework. In the MCE framework, the empirical error rate in training data is approximated by a smoothed and differentiable objective function. Then, an optimization method is used to minimize the objective function with respect to all HMM parameters, such as the generalized probabilistic descend (GPD) algorithm based on gradient descent [19], the approximate second-order Quickprop method [33], a similar EBW-like method [13], and others. Although both MMI and MCE criteria have been shown to be an asymptotic upper bound of the Bayes error when an infinite amount of training data is available [34], a low classification error rate in a *finite* training set does not necessarily guarantee lower error rate in a new test. In practice, several techniques have been used to improve generalization of a discriminative training method, e.g., smoothing sigmoid function in MCE [19], acoustic scaling and weaken language modeling in MMI [45], and so on. With help of these techniques, it has been shown that the discriminative training can significantly improve recognition performance in many very large-scale recognition tasks [8], [26], [28].

On the other hand, the generalization problem of a learning algorithm has been theoretically studied in the field of machine learning, where the Bayes error is shown to be bounded by the classification error rate in a *finite* training set plus a quantity related to the so-called *VC dimension* [35], [39], [44]. The fact that it is the margin[1] in classification rather than the raw training error that matters has become a key tool in recent years when dealing with classifiers. The concept of large margin has been identified as a unifying principle for analyzing many different approaches in pattern classification. [39] As one of the

Manuscript received September 29, 2005; revised May 12, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andreas Stolcke.

H. Jiang is with the Department of Computer Science and Engineering, York University, Toronto, ON M3J 1P3, Canada (e-mail: hj@cs.yorku.ca).

X. Li was with the Department of Computer Science and Engineering, York University, Toronto, ON M3J 1P3, Canada. He is now with Nuance Communications, Burlington, MA 01803 USA (e-mail: xwli@cs.yorku.ca).

C. Liu was with the Department of Computer Science and Engineering, York University, Toronto, ON M3J 1P3, Canada. He is now with the Panasonic San Jose Laboratory, Panasonic R&D Company of America, San Jose, CA 95128-2545 USA (e-mail: chaojunl@research.panasonic.com).

Digital Object Identifier 10.1109/TASL.2006.879805

[1]It has been shown that margin is related to the VC dimension.

most successful examples, support vector machine (SVM) has achieved a huge success in a variety of applications. In the field of speech recognition, we have also observed some research activities over the past years regarding application of SVM to a variety of acoustic modeling tasks. In brief, it starts from some early works in which the standard SVM formulation is directly applied into some isolated speech recognition tasks, e.g., phoneme recognition [6], [10], digit recognition [4], [9], distinct feature detection [30], speaker recognition and verification [41], etc. Since speech patterns are generally regarded as highly complicated and not linearly separable, when we apply SVMs to any speech problems, the first issue we must deal with is how to choose an appropriate kernel to map speech patterns into a high dimension space where a linear classifier is suitable. A variety of kernels have been studied for speech patterns, including the general-purpose polynomial kernels, Gaussian radial basis kernel, and tanh sigmoidal kernels [5], [35]; kernels derived from generative models, e.g., fisher kernels [14], [37] and likelihood-ratio score-space kernel [20], [38]; kernels designed especially for speech patterns, such as dynamic time-alignment kernel which is based on a dynamic time warping (DTW) procedure between two speech sequences [36], probabilistic distance kernel which is based on a KL-divergence between two simple models derived from two speech sequences [29], pair HMM kernel [42], etc. Generally speaking, the original formulation of SVM does not suit well to speech recognition tasks in many aspects. First of all, the standard SVM expects a fixed-length feature vector as input while speech pattern is dynamic and it always leads to variable-length features. To cope with this problem, some researchers have proposed to use proper kernels which are able to map a variable-length feature into a fixed-length one, such as in [20], [29], [36]–[38], [42]. Besides, others also proposed to directly convert the variable-length speech feature sequence into a fixed-size one in a preprocessing stage even though information loss usually occurs in such a conversion, such as linear time warping [6], ad-hoc feature sequence resampling [4], [11], etc. Second, the standard SVM is originally formulated for a binary pattern classification problem. Thus, in speech recognition, it is critical to use SVMs in such an efficient way to make it effective to solve the multiclass problem in ASR. The heuristic approach is to build a set of binary SVMs for all possible classes or class pairs based on the *one versus one* or *one versus rest* approach. During the test, the classification is conducted by combining many local decisions made by all of these binary SVMs according to the strategy of *majority-voting* or *winner-take-all*. In addition, some people have also proposed the so-called *k-class* SVM to generalize the SVM formulation to accommodate multiclass problems, as in [2], [7], [43]. In the *k-class* SVM, margin and objective function are redefined for multiple classes and multiclass classification is solved in a single quadratic optimization. Third, SVM is a static classifier in nature and it is not straightforward to solve sequence recognition problem where the boundary information about each potential pattern is unknown, as in continuous speech recognition. To deal with the dynamic sequence problem in ASR, the first solution is to combine SVM with the existing HMM formulation as in [11], [12], [40]. In these works, a hybrid SVM/HMM formulation is proposed by replacing Gaussian mixture model in each HMM state with an SVM. The raw scores from the SVMs are first converted into probability-like measures by a sigmoid function and then used for HMM likelihood computation. More recently, a novel approach, the so-called *Hidden Markov Support Vector machines (HMSVM)* in [1], is proposed to combine discrete density HMMs (DDHMMs) with SVM for solving the label sequence learning problem in text processing. In HMSVM, DDHMMs are estimated based on the large margin principle just like SVMs. As shown in [1], estimation of DDHMMs for large margin turns out to be a quadratic programming problem under some linear constraints. The problem can be solved by many standard optimization software tools similarly as the standard SVM.

Obviously, the large margin classifiers do provide the theoretical beauty and practical superiority in many applications. However, in speech recognition, Gaussian mixture continuous density HMM (CDHMM) remains the most popular and successful model for modeling speech patterns. As shown in the previous works mentioned previously, direct use of SVMs or loose coupling of SVM with HMM for speech recognition can not easily handle the dynamic nature of speech patterns and will raise technical difficulties in terms of training and recognition complexity which eventually question the feasibility of extending such a framework to some larger scale speech recognition tasks, such as in [8], [17], and [45], which can be easily and efficiently dealt with under the current framework of HMM. Therefore, in this paper, instead of completely switching paradigm from HMM to SVMs or loosely coupling SVM with HMM, we study how to directly estimate Gaussian mixture continuous density HMMs (CDHMMs) for speech recognition based on the large margin principle. In other words, we attempt to estimate the CDHMM parameters in such a way that the decision boundary determined by the estimated CDHMMs achieves the maximum classification margin as in SVMs. An intuitive explanation can be illustrated by a simple HMM-based classifier for a 2-class problem, as shown in Fig. 1. By modifying the HMM parameters, we change the classification boundary to make it as far from all training samples as possible. In this way, margin of the classifier will be increased so that its generalization power is improved accordingly. In this paper, we will show that this type of large margin HMM estimation problem can be formulated as a constrained minimax optimization problem. And, we propose to solve this constrained minimax optimization problem by using a penalized gradient descent algorithm, where the original objective function, i.e., minimum margin, is approximated by a differentiable function, and the constraints are cast as penalty terms in the objective function. The new training method is evaluated in the speaker-independent isolated E-set recognition and the TIDIGITS connected digit string recognition tasks. Experimental results clearly show that the large margin HMMs consistently outperform the conventional HMM training methods, such as ML. It has been consistently observed that the large margin training method yields significant recognition error rate reduction even on top of popular discriminative training methods, such as MCE.

The remainder of this paper is organized as follows. First, we present the general framework for large margin HMM estima-

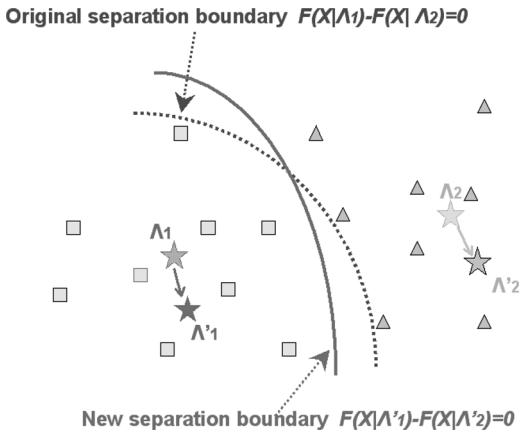


Fig. 1. Illustration of a simple large margin HMM-based classifier.

tion in speech recognition in Section II. Then, in Section III, we give the formulation of large margin estimation for Gaussian mixture CDHMM and mathematically analyze the definition of margin and introduce some theoretically-sound constraints to guarantee the boundedness of the margin in CDHMM. Next, in Section IV, we present a gradient descent based optimization method to approximately solve the constrained minimax optimization in large margin estimation of CDHMM. Then, experimental results on ISOLET isolated E-set recognition and TIDIGITS connected digit string recognition tasks are reported and discussed in Section V. Finally, we conclude the paper with our findings in Section VI.

II. Large Margin HMMs for ASR

In ASR, given any speech utterance X, a speech recognizer will choose the word $\hat{W}$[2] as output based on the plug-in MAP decision rule [15] as follows:

$$\begin{aligned}\hat{W} &= \arg\max_W p(W|X) = \arg\max_W p(W)\cdot p(X|W)\\ &= \arg\max_W p(W)\cdot p(X|\lambda_W) = \arg\max_W \mathcal{F}(X|\lambda_W) \quad (1)\end{aligned}$$

where λ_W denotes the HMM representing the word W and $\mathcal{F}(X|\lambda_W) = p(W)\cdot p(X|\lambda_W)$ is called its discriminant function. In this paper, we are only interested in how to estimate HMM λ_W and assume language model used to calculate $p(W)$ is fixed.

For a speech utterance X_i, assuming its true word identity as W_i, following [1], [7], [43], the multiclass separation margin for X_i is defined as

$$d(X_i) = \mathcal{F}\left(X_i|\lambda_{W_i}\right) - \max_{W_j\in\Omega W_j\neq W_i} \mathcal{F}\left(X_i|\lambda_{W_j}\right) \quad (2)$$

where Ω denotes the set of all possible words. Clearly, the above (2) can be re-arranged into

$$d(X_i) = \min_{W_j\in\Omega W_j\neq W_i} \left[\mathcal{F}\left(X_i|\lambda_{W_i}\right) - \mathcal{F}\left(X_i|\lambda_{W_j}\right)\right] \quad (3)$$

Obviously, if $d(X_i)\leq 0$, X_i will be incorrectly recognized by the current HMM set, denoted as Λ; if $d(X_i) > 0$, X_i will be correctly recognized by the model set Λ.

[2]Depending on the problem of interest, a word W may be any linguistic unit, e.g., a phoneme, a syllable, a word, a phrase, a sentence, etc.

Given a set of training data $\mathcal{D} = \{X_1, X_2, \ldots, X_T\}$, we usually know the true word identities for all utterances in $\mathcal{D}$, denoted as $\mathcal{L} = \{W_1, W_2, \ldots, W_T\}$. Thus, we can calculate the separation margin (or margin for short hereafter) for every utterance in $\mathcal{D}$ based on the definition in (2) or (3). According to the statistical learning theory [44], the generalization error rate of a classifier in new test sets is theoretically bounded by a quantity related to its margin. A large margin classifier usually yields low error rate in new test sets and it shows more robust and better generalization capability. Motivated by the large margin principle, even for those utterances in the training set which all have positive margin, we may still want to maximize the minimum margin to build an HMM-based large margin classifier for ASR. In this paper, we will study how to estimate HMMs for speech recognition based on the principle of maximizing minimum margin.

First, from all utterances in $\mathcal{D}$, we need to identify a subset of utterances $\mathcal{S}$ as

$$\mathcal{S} = \{X_i | X_i\in\mathcal{D} \text{ and } 0\leq d(X_i)\leq\epsilon\} \quad (4)$$

where $\epsilon > 0$ is a preset positive number. Analogically, we call $\mathcal{S}$ as *support vector set* and each utterance in $\mathcal{S}$ is called a support token which has relatively small positive margin among all utterances in the training set $\mathcal{D}$. In other words, all utterances in $\mathcal{S}$ are relatively close to the classification boundary even though all of them locate in the right decision regions. To achieve better generalization power, it is desirable to adjust decision boundaries, which are implicitly determined by all models, through optimizing HMM parameters Λ to make all support tokens as far from the decision boundaries as possible, which will result in a robust classifier with better generalization capability, as shown in Fig. 1. This idea leads to estimating the HMM models Λ based on the criterion of maximizing the minimum margin of all support tokens, which is named as large margin estimation (LME) of HMM

$$\tilde{\Lambda} = \arg\max_\Lambda \min_{X_i\in\mathcal{S}} d(X_i). \quad (5)$$

The HMM models, $\tilde{\Lambda}$, estimated in this way, are called large margin HMMs.

Considering (3), large margin HMMs can be equivalently estimated as follows:

$$\tilde{\Lambda} = \arg\max_\Lambda \min_{X_i\in\mathcal{S}W_j\in\Omega j\neq i} \left[\mathcal{F}\left(X_i|\lambda_{W_i}\right) - \mathcal{F}\left(X_i|\lambda_{W_j}\right)\right]. \quad (6)$$

Finally, the large margin estimation of HMMs can be converted into a standard minimax optimization problem as

$$\tilde{\Lambda} = \arg\min_\Lambda \max_{X_i\in\mathcal{S}W_j\in\Omega j\neq i} \left[\mathcal{F}\left(X_i|\lambda_{W_j}\right) - \mathcal{F}\left(X_i|\lambda_{W_i}\right)\right]. \quad (7)$$

Note that it is fine to include all training data into the support token set with a large value for ϵ in (4). However, this may significantly increase the computational complexity in the following optimization process, and most of those data with large margin are usually inactive in the optimization toward maximizing the minimum one, especially when a gradual optimization method, such as gradient descent, is used.

III. Formulation of Large Margin Estimation for CDHMM

In this section, let us describe how to formulate the large margin estimation in (7) for Gaussian mixture CDHMMs in speech recognition. At first, we assume each speech unit, e.g., a word W, is modeled by an N-state CDHMM with parameter vector $\lambda = (\pi, A, \theta)$, where π is the initial state distribution, $A = \{a_{ij} | 1 \le i, j \le N\}$ is transition matrix, and θ is parameter vector composed of mixture parameters $\theta_i = \{\omega_{ik}, \mu_{ik}, \Sigma_{ik}\}_{k=1,2,\ldots,K}$ for each state i, where K denotes number of Gaussian mixtures in each state. The state observation pdf is assumed to be a mixture of multivariate Gaussian distribution

$$\begin{aligned} p(\mathbf{x}|\theta_i) &= \sum_{k=1}^{K} \omega_{ik} \cdot \mathcal{N}(\mathbf{x}|\mu_{ik}, \Sigma_{ik}) \\ &= \sum_{k=1}^{K} \omega_{ik} \cdot (2\pi)^{-D/2} |\Sigma_{ik}|^{-1/2} \\ &\quad \times \exp\left[-\frac{1}{2}(x-\mu_{ik})^t \Sigma_{ik}^{-1}(x-\mu_{ik})\right] \end{aligned} \tag{8}$$

where mixture weights ω_{ik}s satisfy the constraint $\sum_{k=1}^{K} \omega_{ik} = 1$. In many cases, we prefer to use multivariate Gaussian distribution with diagonal covariance matrix. Thus, the above state observation pdf is simplified as

$$\begin{aligned} p(\mathbf{x}|\theta_i) &= \sum_{k=1}^{K} \omega_{ik} \mathcal{N}(\mathbf{x}|\mu_{ik}, \Sigma_{ik}) \\ &= \sum_{k=1}^{K} \omega_{ik} \prod_{d=1}^{D} \sqrt{\frac{1}{2\pi\sigma_{ikd}^2}} e^{-\frac{(x_d - \mu_{ikd})^2}{2\sigma_{ikd}^2}}. \end{aligned} \tag{9}$$

A. Discriminant Functions of CDHMMs

Given any speech utterance $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_R\}$, let $\mathbf{s} = \{s_1, s_2, \ldots, s_R\}$ be the unobserved state sequence, and $\mathbf{l} = \{l_1, l_2, \ldots, l_R\}$ be the associated sequence of the unobserved mixture component labels, the discriminant function based on the word model λ_W, $\mathcal{F}(X|\lambda_W)$, can be expressed as

$$\begin{aligned} \mathcal{F}(X|\lambda_W) = \sum_{\mathbf{s}} \sum_{\mathbf{l}} \Big\{ & \pi_{s_1} \omega_{s_1 l_1} \mathcal{N}\left(\mathbf{x}_1 | \mu_{s_1 l_1}, \Sigma_{s_1 l_1}\right) \\ & \times \prod_{t=2}^{R} a_{s_{t-1} s_t} \cdot \omega_{s_t l_t} \cdot \mathcal{N}\left(\mathbf{x}_t | \mu_{s_t l_t}, \Sigma_{s_t l_t}\right) \Big\} \cdot p(W) \end{aligned} \tag{10}$$

where the summations are taken over all possible state and mixture component label sequences. If we adopt the Viterbi method to approximate the above summation with the single optimal Viterbi path, denoted as $\mathbf{s}^* = \{s_1^*, s_2^*, \ldots, s_R^*\}$ and $\mathbf{l}^* = \{l_1^*, l_2^*, \ldots, l_R^*\}$, then we have

$$\begin{aligned} \mathcal{F}(X|\lambda_W) \approx \pi_{s_1^*} \omega_{s_1^* l_1^*} \mathcal{N}\left(\mathbf{x}_1 | \mu_{s_1^* l_1^*}, \Sigma_{s_1^* l_1^*}\right) \prod_{t=2}^{R} a_{s_{t-1}^* s_t^*} \cdot \omega_{s_t^* l_t^*} \\ \cdot \mathcal{N}\left(\mathbf{x}_t | \mu_{s_t^* l_t^*}, \Sigma_{s_t^* l_t^*}\right) \cdot p(W). \end{aligned} \tag{11}$$

Usually, it is more convenient to represent the discriminant function $\mathcal{F}(X_i|\lambda_{W_j})$ in the logarithm scale. Assume we adopt diagonal covariance matrices for all Gaussian mixtures, we have

$$\begin{aligned} & \mathcal{F}(X|\lambda_W) \\ & \approx \log p(W) + \log \pi_{s_1^*} + \sum_{t=2}^{R} \log a_{s_{t-1}^* s_t^*} + \sum_{t=1}^{R} \log \omega_{s_t^* l_t^*} \\ & \quad - \frac{1}{2} \sum_{t=1}^{R} \sum_{d=1}^{D} \left[\log \sigma_{s_t^* l_t^* d}^2 + \frac{\left(x_{itd} - \mu_{s_t^* l_t^* d}\right)^2}{\sigma_{s_t^* l_t^* d}^2} \right]. \end{aligned} \tag{12}$$

Throughout this paper, for simplicity, we only consider to estimate mean vectors of CDHMMs based on the large margin principle while keeping all other CDHMM parameters constant during the large margin estimation. For any utterance X_i in the support token set $\mathcal{S}$, if we assume its true model is λ_{W_i}, then we check for all other models $\lambda_{W_j} (j \ne i)$ to include those hypothesized incorrect model λ_{W_j} in the optimization procedure for large margin model estimation as long as they meet the condition $0 < \mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_{W_j}) \le \epsilon$, where ϵ is a preset threshold. For simplicity, we use the Viterbi approximation in evaluating both $\mathcal{F}(X_i|\lambda_{W_i})$ and $\mathcal{F}(X_i|\lambda_{W_j})$. For $\mathcal{F}(X_i|\lambda_{W_i})$, let us assume the optimal Viterbi path is $\mathbf{s}_i^* = \{s_{i1}^*, s_{i2}^*, \ldots, s_{iT}^*\}$ and $\mathbf{l}_i^* = \{l_{i1}^*, l_{i2}^*, \ldots, l_{iT}^*\}$. Similarly, we assume the optimal path is $\mathbf{s}_j^* = \{s_{j1}^*, s_{j2}^*, \ldots, s_{jT}^*\}$ and $\mathbf{l}_j^* = \{l_{j1}^*, l_{j2}^*, \ldots, l_{jT}^*\}$ when evaluating $\mathcal{F}(X_i|\lambda_{W_j})$. Since we are only considering to estimate mean vectors of CDHMMs, we can rewrite $\mathcal{F}(X_i|\lambda_{W_i})$ and $\mathcal{F}(X_i|\lambda_{W_j})$ according to (12) as follows:

$$\mathcal{F}\left(X_i|\lambda_{W_i}\right) \approx C_i' - \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} \frac{\left(x_{itd} - \mu_{s_{it}^* l_{it}^* d}\right)^2}{\sigma_{s_{it}^* l_{it}^* d}^2} \tag{13}$$

$$\mathcal{F}\left(X_i|\lambda_{W_j}\right) \approx C_j'' - \frac{1}{2} \sum_{t=1}^{T} \sum_{d=1}^{D} \frac{\left(x_{itd} - \mu_{s_{jt}^* l_{jt}^* d}\right)^2}{\sigma_{s_{jt}^* l_{jt}^* d}^2} \tag{14}$$

where C_i' and C_j'' are two constants independent from mean vectors. In this case, the discriminant functions $\mathcal{F}(X_i|\lambda_{W_i})$ and $\mathcal{F}(X_i|\lambda_{W_j})$ can be represented as a summation of some quadratic terms related to mean values of CDHMMs.

B. Imposing Constraints for CDHMM in Large Margin Estimation

As shown in [22] and [24], the decision margins, $d_{ij}(X_i) = \mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_{W_j})$ as in (3), are actually unbounded for the CDHMMs, which in turn makes the margin as defined in (2) unbounded for CDHMMs as well. In other words, we can adjust CDHMM parameters in a way to increase the margin unlimitedly so that the minimax optimization in (7) is actually not solvable if not imposing any other constraints in optimization. Several methods have been proposed to solve this problem to make the margin bounded. In [22], a heuristic method, called *Iterative Localized Optimization*, is used to guarantee the existence of an optimal point. In that method, instead of optimizing parameters of all models at the same time, only one selected model will be adjusted in each step of optimization, then the process iterates to update another model until the optimal margin is achieved. In

[24] and [25], we replace the original definition of margin by a relative separation margin which is bounded by definition.

In this paper, we will mathematically analyze the definition of margin and introduce some theoretically sound constraints for the minimax optimization in LME of CDHMMs in speech recognition. First of all, if we adopt the Viterbi approximation as above, the discriminant functions $\mathcal{F}(X_i|\lambda_{W_i})$ and $\mathcal{F}(X_i|\lambda_{W_j})$ in (13) and (14) can be represented as a summation of quadratic terms of CDHMMs' mean vectors. As a result, the decision margin $d_{ij}(X_i)$ can be represented as a standard diagonal quadratic form

$$\begin{aligned} d_{ij}(X_i) &= \mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}\left(X_i|\lambda_{W_j}\right) \\ &\approx C_{ij} - \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}\left[\frac{\left(x_{itd}-\mu_{s_{it}^* l_{it}^* d}\right)^2}{\sigma^2_{s_{it}^* l_{it}^* d}} - \frac{\left(x_{itd}-\mu_{s_{jt}^* l_{jt}^* d}\right)^2}{\sigma^2_{s_{jt}^* l_{jt}^* d}}\right] \end{aligned} \quad (15)$$

where $C_{ij} = C_i' - C_j''$.

Clearly, from (15) we can see that each feature dimension x_{itd} contributes to the above decision margin separately. Thus, for each feature vector $\mathbf{x}_{it}$, we can divide all of its dimensions into two parts: $D_{t1} = \{d|\sigma^2_{s_{it}^* l_{it}^* d} = \sigma^2_{s_{jt}^* l_{jt}^* d}\}$ and $D_{t2} = \{d|\sigma^2_{s_{it}^* l_{it}^* d} \neq \sigma^2_{s_{jt}^* l_{jt}^* d}\}$. After some math manipulation, we have

$$\begin{aligned} d_{ij}(X_i) \approx C_{ij} + \sum_{t=1}^{R}\Bigg\{ &\sum_{d\in D_{t1}} \mathcal{K}_{itd}\cdot(x_{itd}-\mathcal{L}_{itd}) \\ &+ \sum_{d\in D_{t2}}\left[\mathcal{A}_{itd}\cdot(x_{itd}-\mathcal{B}_{itd})^2 - \mathcal{C}_{itd}\right]\Bigg\} \end{aligned} \quad (16)$$

where

$$\mathcal{K}_{itd} = \frac{\mu_{s_{it}^* l_{it}^* d} - \mu_{s_{jt}^* l_{jt}^* d}}{\sigma_{s_{it}^* l_{it}^* d}\cdot\sigma_{s_{jt}^* l_{jt}^* d}} \quad (17)$$

$$\mathcal{L}_{itd} = \frac{\mu_{s_{it}^* l_{it}^* d} + \mu_{s_{jt}^* l_{jt}^* d}}{2} \quad (18)$$

$$\mathcal{A}_{itd} = \frac{\sigma^2_{s_{it}^* l_{it}^* d} - \sigma^2_{s_{jt}^* l_{jt}^* d}}{2\sigma^2_{s_{it}^* l_{it}^* d}\sigma^2_{s_{jt}^* l_{jt}^* d}} \quad (19)$$

$$\mathcal{B}_{itd} = \frac{\mu_{s_{jt}^* l_{jt}^* d}\cdot\sigma^2_{s_{it}^* l_{it}^* d} - \mu_{s_{it}^* l_{it}^* d}\cdot\sigma^2_{s_{jt}^* l_{jt}^* d}}{\sigma^2_{s_{it}^* l_{it}^* d} - \sigma^2_{s_{jt}^* l_{jt}^* d}} \quad (20)$$

$$\mathcal{C}_{itd} = \frac{\left(\mu_{s_{it}^* l_{it}^* d} - \mu_{s_{jt}^* l_{jt}^* d}\right)^2}{2\left(\sigma^2_{s_{it}^* l_{it}^* d} - \sigma^2_{s_{jt}^* l_{jt}^* d}\right)}. \quad (21)$$

From (16), we can see that for those dimensions where the two models, i.e., λ_{W_i} and λ_{W_j}, have the same variance, its margin contribution degenerates to a linear function of x_{itd}. And, for those dimensions where two models have different variance, its margin contribution is a quadratic function of x_{itd}. For the linear part, it is clear that we can increase the slope value, namely $\mathcal{K}_{itd}$, to increase its margin contribution unlimitedly for any given data sample x_{itd}. Similarly, for the quadratic part, we can move the position of parabola vertex $\mathcal{B}_{itd}$ toward infinity to increase its margin contribution as much as we want. Therefore, the decision margin of CDHMMs $d_{ij}(X_i)$ is actually unbounded for any given data X_i, which in turn makes the margin as defined in (2) unbounded for CDHMMs as well. In other words, for any given speech data X_i, we can adjust CDHMM parameters, i.e., Gaussian means in this study, in such a way to increase the margin unlimitedly so that the minimax optimization in (5) is actually not solvable unless we impose additional constraints with respect to CDHMM model parameters during optimization.

According to the previous analysis, in order to bound the margin contribution from the linear part, intuitively we should constrain the norm of slope vector in all linear dimensions D_{t1} to be a finite value. As a result, we should add the following constraint during the optimization:

$$R_1\left(\lambda_{W_i},\lambda_{W_j}|X_i\right) = \sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{itd}^2 = g_{ij}^2 \quad (22)$$

where g_{ij} is a constant set to the value of R_1 calculated based on initial models.

On the other hand, in order to bound the margin contribution from the quadratic part, intuitively we should limit the position of parabola vertex by adding the following spherical constraint:

$$R_2\left(\lambda_{W_i},\lambda_{W_j}|X_i\right) = \sum_{t=1}^{R}\sum_{d\in D_{t2}}\left(\mathcal{B}_{itd} - \mathcal{B}_{itd}^{(0)}\right)^2 \le G_{ij}^2 \quad (23)$$

where G_{ij} is another preset constant and $\mathcal{B}_{itd}^{(0)}$ is also a constant which is set to be the value of $\mathcal{B}_{itd}$ computed based on the initial models. Actually, we have the following theorem regarding the boundedness of the margin $d(X_i)$.

Theorem III.1: Assume we have a set of CDHMMs, $\Lambda = \{\lambda_1,\lambda_2,\ldots,\lambda_K\}$ and a set of training data, denoted as $\mathcal{D} = \{X_1,X_2,\ldots,X_N\}$. The margin $d(X_i)$, as defined in (2), is bounded for any token X_i in the training set $\mathcal{D}$ as long as the following constraints satisfy. 1) The constraints in (22) and (23) holds simultaneously between any two models, λ_i and λ_j, in Λ. 2) For any model λ_i in Λ, there is at least one token in $\mathcal{D}$ which belongs to this model and is correctly classified by the current model set Λ, i.e., $d(X_i) > 0$ holds.

Proof: Since we have $d(X_i) = \min_{j\neq i} d_{ij}(X_i)$, we can prove $d(X_i)$ is bounded as long as we prove $d_{ij}(X_i)$ is bounded for any two models, λ_i and λ_j in Λ.

We can rewrite $d_{ij}(X_i)$ in (16) into

$$\begin{aligned} d_{ij}(X_i) = &\underbrace{\sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{itd}\cdot x_{itd}}_{\text{Item I}} \\ &+ \underbrace{\sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{A}_{itd}\cdot(x_{itd}-\mathcal{B}_{itd})^2}_{\text{Item II}} - \Gamma_{ij} \end{aligned} \quad (24)$$

where

$$\Gamma_{ij} = \sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{itd}\cdot\mathcal{L}_{itd} + \sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{C}_{itd} - C_{ij}. \quad (25)$$

Based on Cauchy–Schwarz inequality, we can get

$$\left|\sum_{t=1}^{R}\sum_{d\in D_1}\mathcal{K}_{itd}\cdot x_{itd}\right| \le \sqrt{\sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{itd}^2}\sqrt{\sum_{t=1}^{R}\sum_{d\in D_1}x_{itd}^2} = g_{ij}\sqrt{\sum_{t=1}^{R}\sum_{d\in D_{t1}}x_{itd}^2} = g_{ij}\cdot B_1(X_i). \tag{26}$$

Since the training set $\mathcal{D}$ is given and fixed, $B_1(X_i)$ is a finite value. In this way, we have proved that Item I in (24) is bounded.

For any two models λ_i and λ_j, in Λ, since the constraint in (23) holds, for any component $\mathcal{B}_{itd}$, we have

$$\left(\mathcal{B}_{itd}-\mathcal{B}_{itd}^{(0)}\right)^2 \le \sum_{t=1}^{R}\sum_{d\in D_{t2}}\left(\mathcal{B}_{itd}-\mathcal{B}_{itd}^{(0)}\right)^2 \le G_{ij}^2. \tag{27}$$

Thus

$$\mathcal{B}_{itd}^{(0)}-G_{ij}\le\mathcal{B}_{itd}\le\mathcal{B}_{itd}^{(0)}+G_{ij}. \tag{28}$$

Then we have

$$\begin{aligned}&\left|\sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{A}_{itd}\cdot(x_{itd}-\mathcal{B}_{itd})^2\right|\\ &\le\sum_{t=1}^{R}\sum_{d\in D_{t2}}|\mathcal{A}_{itd}|(x_{itd}-\mathcal{B}_{itd})^2\\ &\le\sum_{t=1}^{R}\sum_{d\in D_2}|\mathcal{A}_{itd}|\\ &\cdot\max\left\{\left(x_{itd}-\mathcal{B}_{itd}^{(0)}+G_{ij}\right)^2,\left(x_{itd}-\mathcal{B}_{itd}^{(0)}-G_{ij}\right)^2\right\}\\ &\le\sum_{t=1}^{R}\sum_{d\in D_2}|\mathcal{A}_{itd}|\cdot D_{itd}(X_i).\end{aligned} \tag{29}$$

In this study, we only consider to estimate Gaussian mean vectors in CDHMMs, thus $\mathcal{A}_{itd}$ is constant. And $D_{itd}(X_i)$ is also a finite value since the training set $\mathcal{D}$ is fixed during the optimization. Therefore, we have proven that Item II in (24) is also bounded.

Next, we will prove that Γ_{ij} in (24) is bounded as well. For any model in Λ, we have at least one token which has positive margin. Here, let us assume that for model λ_i, the token X_i has positive margin, i.e., $d(X_i)>0$, and for model λ_j, the token X_j has positive margin, $d(X_j)>0$. As a result, we have

$$\begin{aligned}d_{ij}(X_i)&=\mathcal{F}(X_i|\lambda_i)-\mathcal{F}(X_i|\lambda_j)\\ &=\sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{itd}\cdot x_{itd}\\ &\quad+\sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{A}_{itd}(x_{itd}-\mathcal{B}_{itd})^2-\Gamma_{ij}>0\end{aligned} \tag{30}$$

$$\begin{aligned}d_{ji}(X_j)&=\mathcal{F}(X_j|\lambda_j)-\mathcal{F}(X_j|\lambda_i)\\ &=\sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{jtd}\cdot x_{jtd}\\ &\quad+\sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{A}_{jtd}(x_{jtd}-\mathcal{B}_{jtd})^2-\Gamma_{ji}>0.\end{aligned} \tag{31}$$

From (30), we have

$$\Gamma_{ij}<\underbrace{\sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{itd}\cdot x_{itd}}_{\text{Item I}}+\underbrace{\sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{A}_{itd}(x_{itd}-\mathcal{B}_{itd})^2}_{\text{Item II}}. \tag{32}$$

Based on the above analysis, both Item I and Item II in (32) are bounded. Thus, Γ_{ij} is upper bounded.

Based on Γ_{ij} defined in (25) as well as (17), (18), and (21), it is clear that $\Gamma_{ij}=-\Gamma_{ji}$. Thus, according to (31), we have

$$\Gamma_{ij}>-\underbrace{\sum_{t=1}^{R}\sum_{d\in D_{t1}}\mathcal{K}_{jtd}\cdot x_{jtd}}_{\text{Item III}}-\underbrace{\sum_{t=1}^{R}\sum_{d\in D_{t2}}\mathcal{A}_{jtd}(x_{jtd}-\mathcal{B}_{jtd})^2}_{\text{Item IV}}. \tag{33}$$

Similarly, since both Item III and Item IV are bounded, Γ_{ij} is lower bounded.

Finally, based on (24), $d_{ij}(X_i)$ is bounded for any two models, λ_i and λ_j. Thus, $d(X_i)$ is bounded. ■

According to Theorem III.1, the minimum margin in (7) is a bounded function of model parameter set Λ under the conditions specified in Theorem III.1. Thus, we can always search for an appropriate set of model parameter to maximize the minimum margin. Therefore, the minimax optimization problem in (5) becomes solvable under these constraints. Here, we reformulate the large margin estimation as the following constrained minimax optimization problem:

$$\tilde{\Lambda}=\arg\min_{\Lambda}\max_{X_i\in\mathcal{S}}\max_{W_j\in\Omega\, i\neq j}\left[\mathcal{F}\left(X_i|\lambda_{W_j}\right)-\mathcal{F}\left(X_i|\lambda_{W_i}\right)\right] \tag{34}$$

subject to

$$R_1\left(\lambda_{W_i},\lambda_{W_j}|X_i\right)=g_{ij}^2 \tag{35}$$

$$R_2\left(\lambda_{W_i},\lambda_{W_j}|X_i\right)\le G_{ij}^2 \tag{36}$$

$$\mathcal{F}\left(X_i|\lambda_{W_j}\right)-\mathcal{F}\left(X_i|\lambda_{W_i}\right)<0 \tag{37}$$

for all $X_i\in\mathcal{S}$ and $W_j\in\Omega$ and $W_j\neq W_i$. Here, g_{ij} and G_{ij} are preset constants calculated according to the decoding sequences of X_i based on the initial models of W_i and W_j. Note that we do not explicitly impose condition 2) of Theorem III.1 during the previous optimization. As long as we choose a reasonably large set of support tokens $\mathcal{S}$, in practice we usually have at least one support token to limit each model parameter to make sure that Γ_{ij} is bounded for any two model λ_i and λ_j. Besides, we introduce constraints in (37) to ensure that none of these support tokens will cross decision boundary to have negative margin during optimization.

IV. Optimization Based on the Penalized Gradient Descent

As shown in previously, large margin estimation (LME) of CDHMMs turns out to be a constrained minimax optimization as shown in (34). Obviously, it is a complicated nonlinear optimization problem, where typically no efficient solution exists from the viewpoint of optimization theory. In practice, this kind of minimax optimization can be numerically solved with some general-purpose optimization software package. However, due to the huge number of free parameters in any CDHMM-based speech recognition system, it is very difficult, if not impossible, to directly use any general-purpose optimization tool to solve the above constrained minimax optimization problem in an efficient way. In this paper, we propose to approximate the objective function in the above minimax optimization with a differentiable function and then derive an iterative optimization approach for CDHMM mean vectors based on the gradient descent method. As opposed to *Iterative Localized Optimization* method in [22], we call it *Constrained Joint Optimization* method in this paper.

A. Iterative Optimization Based on Gradient Descent

To construct a differentiable objective function for the large margin optimization in (34), we first need to approximate *max* operation with a continuous and differentiable function. A well-known trick is to use the summation of exponential functions, as used in the MCE [19]. That is, the maximization in (34) is approximated by summation of exponential functions as follows:

$$\max_{X_i\in\mathcal{S}}\max_{W_j\in\Omega\, j\neq i}\left[\mathcal{F}\left(X_i|\lambda_{W_j}\right)-\mathcal{F}\left(X_i|\lambda_{W_i}\right)\right] \approx \log\left[\sum_{X_i\in\mathcal{S}}\sum_{W_j\in\Omega\, j\neq i}\exp\left[\eta\cdot\mathcal{F}\left(X_i|\lambda_{W_j}\right)-\eta\cdot\mathcal{F}\left(X_i|\lambda_{W_i}\right)\right]\right]^{1/\eta} \quad (38)$$

where $\eta > 1$. As $\eta \to \infty$, the continuous function in the right-hand side of (38) will approach the maximization in the left-hand side. In practice, we can choose η as a constant significantly larger than 1.

From (38), we construct the objective function, called *smoothed margin*, for large margin estimation (LME) of CDHMM as follows:

$$Q(\Lambda)=\frac{1}{\eta}\log\left[\sum_{X_i\in\mathcal{S}}\sum_{W_j\in\Omega\, j\neq i}\exp\left[\eta\cdot\mathcal{F}\left(X_i|\lambda_{W_j}\right)-\eta\cdot\mathcal{F}\left(X_i|\lambda_{W_i}\right)\right]\right] =\frac{1}{\eta}\log\left[\sum_{X_i\in\mathcal{S}}\sum_{W_j\in\Omega\, j\neq i}\exp\left[-\eta\cdot d_{ij}(X_i)\right]\right] \quad (39)$$

where $\mathcal{F}(X_i|\lambda_{W_j})$ is calculated as in (10) or (12). Then, an iterative gradient descent method must be used to minimize $Q(\Lambda)$ with respect to all CDHMM mean vectors Λ to approximately derive the large margin estimation of CDHMM as originally defined in (34). The minimization is subject to all constraints given in (35) to (37). In practice, the constrained minimization problem can be transformed into an unconstrained minimization problem by casting all constraints[3] as penalty terms in the objective function

$$\begin{aligned}O(\Lambda)&=Q(\Lambda)+\tau_1\cdot\sum_{X_i\in\mathcal{S}}\sum_{j\neq i}\left(R_1\left(\lambda_{W_j},\lambda_{W_i}|X_i\right)-g_{ij}\right)^2\tau_2\\&\quad\cdot\sum_{X_i\in\mathcal{S}}\sum_{j\neq i}\max\left\{0,R_2\left(\lambda_{W_j},\lambda_{W_i}|X_i\right)-G_{ij}^2\right\}^2\\&=Q(\Lambda)+\tau_1\cdot P_1(\Lambda)+\tau_2\cdot P_2(\Lambda)\end{aligned} \quad (40)$$

where τ_1 and τ_2 are two large positive numbers to balance the penalty terms, and we define

$$P_1(\Lambda)=\sum_{X_i\in\mathcal{S}}\sum_{j\neq i}\left(R_1\left(\lambda_{W_j},\lambda_{W_i}|X_i\right)-g_{ij}\right)^2 \quad (41)$$

$$P_2(\Lambda)=\sum_{X_i\in\mathcal{S}}\sum_{j\neq i}\max\left\{0,R_2\left(\lambda_{W_j},\lambda_{W_i}|X_i\right)-G_{ij}^2\right\}^2. \quad (42)$$

Following [18], we introduce the following transformation to normalize mean vectors during the model estimation process:

$$\tilde{\mu}_{skl}^m=\mu_{skl}^m/\sigma_{skl}^m \quad (43)$$

where $\tilde{\mu}_{skl}^m$ is the transformed parameter of lth dimension of Gaussian mean vector for the kth mixture component of state s of HMM model λ_m. And the gradient descent algorithm is used to adjust Gaussian means to minimize the objective function $O(\Lambda)$ as follows:

$$\tilde{\mu}_{skl}^m(n+1)=\tilde{\mu}_{skl}^m(n)-\epsilon\cdot\left.\frac{\partial O(\Lambda)}{\partial\tilde{\mu}_{skl}^m}\right|_{\Lambda=\Lambda_n} \quad (44)$$

$$\mu_{skl}^m(n+1)=\tilde{\mu}_{skl}^m(n+1)\sigma_{skl}^m \quad (45)$$

where ϵ is the step size, and $\tilde{\mu}_{skl}^m(n+1)$ denotes normalized mean of lth dimension of Gaussian mean vector for the kth mixture component of state s of HMM model λ_m at $(n+1)$th iteration and $\mu_{skl}^m(n+1)$ its counterpart in original model space.

Furthermore, we have

$$\frac{\partial O(\Lambda)}{\partial\tilde{\mu}_{skl}^m}=\frac{\partial Q(\Lambda)}{\partial\tilde{\mu}_{skl}^m}+\tau_1\cdot\frac{\partial P_1(\Lambda)}{\partial\tilde{\mu}_{skl}^m}+\tau_2\cdot\frac{\partial P_2(\Lambda)}{\partial\tilde{\mu}_{skl}^m}. \quad (46)$$

From (39), we have

$$\frac{\partial Q(\Lambda)}{\partial\tilde{\mu}_{skd}^m}=\frac{1}{\eta}\frac{1}{Q_1}\frac{\partial Q_1(\Lambda)}{\partial\tilde{\mu}_{skd}^m} \quad (47)$$

where we denote

$$Q_1(\Lambda)=\sum_{X_i\in\mathcal{S}}\sum_{j\neq i}\exp\left[\eta\cdot d_{ij}(X_i)\right]. \quad (48)$$

And we have

$$\frac{\partial Q_1}{\partial\tilde{\mu}_{skl}^m}=\sum_{X_i\in\mathcal{S}}\sum_{j\neq i}\frac{\partial\left[\exp\left(\eta\cdot d_{ij}(X_i)\right)\right]}{\partial\tilde{\mu}_{skl}^m} \quad (49)$$

[3]The constraint (37) is not explicitly included since we use gradient descent to maximize margin in the support token set. Initially, the constraint (37) holds since minimum margin is positive in support token set. In each step of gradient descent, the minimum margin is gradually increased. Thus, the constraint (37) is not actually active in this type of gradient descent optimization as long as step size is small enough.

$$= \sum_{X_i \in \mathcal{S}} \Big\{ -\delta(W_i - m) \frac{\partial F(X_i|\lambda_m)}{\partial \tilde{\mu}^m_{skl}} \times \sum_{W_j \in \Omega} \sum_{j \neq i} \eta \cdot \exp\left[\eta \cdot d_{ij}(X_i)\right] + (1 - \delta(W_i - m))\, \eta \exp\left[\eta \cdot d_{ij}(X_i)\right] \times \frac{\partial F(X_i|\lambda_m)}{\partial \tilde{\mu}^m_{skl}} \Big\} \quad (50)$$

where $\delta(\cdot)$ stands for the Dirac delta function

$$\frac{\partial \mathcal{F}(X_i|\lambda_m)}{\partial \tilde{\mu}^m_{skl}} = \sum_{t=1}^{T} \delta(s_t - s)\delta(l_t - k) \frac{x_{itl} - \mu^m_{skl}}{\sigma^m_{skl}}, \quad (51)$$

Besides, we have

$$\frac{\partial P_1}{\partial \tilde{\mu}^m_{skl}} = \sum_{X_i \in \mathcal{S}} \sum_{W_j \in \Omega} \sum_{j \neq i} \frac{\partial \left(R_1\left(\lambda_{W_j}, \lambda_{W_i}|X_i\right) - g_{ij}\right)^2}{\partial \tilde{\mu}^m_{skl}}$$
$$= 2 \sum_{X_i \in \mathcal{S}} \Big\{ \delta(W_i - m) \sum_{W_j \in \Omega} \sum_{j \neq i} \left(R_1\left(\lambda_{W_j}, \lambda_m|X_i\right) - g_{ij}\right) \times \frac{\partial R_1\left(\lambda_{W_j}, \lambda_m|X_i\right)}{\partial \tilde{\mu}^m_{skl}} + (1 - \delta(W_i - m)) \times \left(R_1(\lambda_m, \lambda_{W_i}|X_i) - g_{ij}\right) \times \frac{\partial R_1\left(\lambda_m, \lambda_{W_i}|X_i\right)}{\partial \tilde{\mu}^m_{skl}} \Big\} \quad (52)$$

since

$$R_1\left(\lambda_{W_j}, \lambda_m|X_i\right) = \sum_{t=1}^{T} \sum_{d \in D_{t1}} \frac{\left(\mu^m_{s^m_t l^m_t d} - \mu^j_{s^j_t l^j_t d}\right)^2}{\left(\sigma^m_{s^m_t l^m_t d} \sigma^j_{s^j_t l^j_t d}\right)^2}. \quad (53)$$

Thus, we have

$$\frac{\partial R_1\left(\lambda_{W_j}, \lambda_m|X_i\right)}{\partial \tilde{\mu}^m_{skl}} = 2 \sum_{t=1}^{T} \frac{\left(\mu^m_{skl} - \mu^j_{s^j_t l^j_t l}\right)}{\sigma^m_{skl} \left(\sigma^j_{s^j_t l^j_t l}\right)^2} \times \delta\left(s^m_t - s\right) \delta\left(l^m_t - k\right) \delta(l \in D_{t1}). \quad (54)$$

Next, we have

$$\frac{\partial P_2}{\partial \tilde{\mu}^m_{skl}} = \sum_{X_i \in \mathcal{S}} \sum_{j \neq i} \frac{\partial \max\left\{0, R_2\left(\lambda_{W_j}, \lambda_{W_i}|X_i\right) - G^2_{ij}\right\}^2}{\partial \tilde{\mu}^m_{skl}}. \quad (55)$$

Here, we define its differential as

$$\frac{\partial \max\left\{0, R_2\left(\lambda_{W_j}, \lambda_{W_i}|X_i\right) - G^2_{ij}\right\}^2}{\partial \tilde{\mu}^m_{skl}} = \begin{cases} \frac{\partial \left(R_2 - G^2_{ij}\right)^2}{\partial \tilde{\mu}^m_{skl}}, & \text{if } R_2 > G^2_{ij} \\ 0, & \text{if } R_2 \leq G^2_{ij} \end{cases} \quad (56)$$

where

$$\frac{\partial \left(R_2 - G^2_{ij}\right)^2}{\partial \tilde{\mu}^m_{skl}} = 2\left(R_2\left(\lambda_{W_j}, \lambda_{W_i}|X_i\right) - G^2_{ij}\right) \times \frac{\partial R_2\left(\lambda_{W_j}, \lambda_{W_i}|X_i\right)}{\partial \tilde{\mu}^m_{skl}}$$
$$= 2\left(R_2\left(\lambda_{W_j}, \lambda_{W_i}|X_i\right) - G^2_{ij}\right) \times \left[\delta(W_i - m) \frac{\partial R_2\left(\lambda_{W_j}, \lambda_m|X_i\right)}{\partial \tilde{\mu}^m_{skl}} + \delta(W_j - m) \frac{\partial R_2\left(\lambda_m, \lambda_{W_i}|X_i\right)}{\partial \tilde{\mu}^m_{skl}}\right] \quad (57)$$

since

$$R_2\left(\lambda_m, \lambda_{W_i}|X_i\right) = \sum_{t=1}^{R} \sum_{d \in D_{t2}} \left(\mathcal{B}_{itd} - \mathcal{B}^{(0)}_{itd}\right)^2. \quad (58)$$

Thus, we have

$$\frac{\partial R_2\left(\lambda_m, \lambda_{W_i}|X_i\right)}{\partial \tilde{\mu}^m_{skl}} = 2 \sum_{t=1}^{T} \left[\left(\mathcal{B}_{itd} - \mathcal{B}^{(0)}_{itd}\right) \frac{\mu_{skl} \cdot \left(\sigma^i_{s^i_t l^i_t d}\right)^2 \cdot \sigma^m_{skl}}{\left(\sigma^i_{s^i_t l^i_t d}\right)^2 - \left(\sigma^m_{skl}\right)^2} \times \delta\left(s^m_t - s\right) \delta\left(l^m_t - k\right) \delta(l \in D_{t2})\right]. \quad (59)$$

As a remark, the whole LME training process is summarized in Algorithm 1. In each epoch, we first recognize all training data based on the current model parameters. Then we select support tokens according to (4) and obtain the optimal Viterbi sequence for each support token according to its recognition result. Then, a penalized gradient descent algorithm is run several iterations to optimize the constrained minimax optimization problem with respect to all Gaussian means jointly. Then, if not convergent, the next epoch will start from decoding all training data again.

Algorithm 1 Constrained Joint Optimization

repeat

1. Perform Viterbi decoding for each utterance in training set based on current models.
2. Identify the support set $\mathcal{S}$ based on the current model set $\Lambda^{(n)}$ according to (4).
3. Obtain optimal Viterbi paths for all support tokens.
4. A number of iterations of gradient descent updates are run to maximize the minimum margin subject to the corresponding constraints w.r.t. Λ : $\Lambda^{(n)} \Rightarrow \Lambda^{(n+1)}$.
5. $n = n + 1$.

until some convergence conditions are met

V. Experiments

A. Isolated Speech Recognition: ISOLET E-set Recognition

In our first set of experiments, the LME training based on the constrained joint optimization method is evaluated on the English E-set recognition with OGI ISOLET database, consisting of {B, C, D, E, G, P, T, V, Z}. ISOLET is a database of letters of the English alphabet spoken in isolation. The database consists of 7800 spoken letters, two productions of each letter by 150 speakers, 75 male and 75 female. The recordings were done under quiet, laboratory conditions with a noise-canceling microphone. The data were sampled at 16 kHz with 16-bit quantization. ISOLET is divided into five parts named ISOLET 1–5. In our experiment, only the first production of each letter in ISOLET 1–4 is used as training data (1080 utterances). All data in ISOLET 5 is used as testing data (540 utterances). The feature vector is of 39 dimensions, which include 12-d static mel frequency cepstral coefficient (MFCC), log-energy, delta, and acceleration coefficients. An HMM recognizer with 16-state whole-word-based models is trained based on different training criteria. Here CDHMMs with 1-mixture, 2-mixture and 4-mixture per state are experimented. In each case, a maximum likelihood estimation (MLE) model is trained based on the standard Baum–Welch algorithm using HTK 3.0. Then, the best MLE model is used as the seed model to conduct minimum classification error (MCE) training. The MCE algorithm with online update is implemented exactly following [18]. The two parameters of the sigmoid function are experimentally tuned for the best performance on the test set: the slope value $\gamma = 2.0$ and shift value $\theta = 0$ are used for this task. All other E-set models are used as competing models with smoothing factor $\eta = 4.0$ in the MCE training. All parameters are tuned based on 1-mixture model and used for all others. Next, we use the best MCE models as the initial models to perform large margin estimation (LME) training. In LME training, we only update Gaussian mean vectors. In each epoch, ϵ in (4) is set to include appropriate number of support tokens and η in (38) is automatically set to make the largest value of $\eta \cdot d_{ij}(X_i)$ around -10.0 to avoid underflow. G_{ij} in (23) is chosen to be 0.01. In this task, recognition errors in training data are quickly brought down to zero after a couple of iterations in MCE training, and training error rate remains zero in LME training of this task. In the ISOLET E-set task, we usually run a large number of iterations of gradient descent updates with a very small step size during each epoch of the *Constrained Joint Optimization* method.

1) Effectiveness of the Constraints: First of all, we study the effect of the constraints (35) to (37) which we introduced for the minimax optimization in this paper. In Fig. 2, we plot the objective function, i.e., $Q(\Lambda)$ in (39), as a function of iteration number in gradient descent updates during the first epoch of the constrained joint optimization method. It is clearly shown that after adding the constraints (35) to (37) into the optimization the objective function converges to a local minimal point after a number of iterations, which shows the effectiveness of the constraints (35) to (37). The learning curves as a function of epochs are shown in Fig. 4

Second, in Fig. 3, we plot number of support tokens as a function of number of epochs when we run the *constrained joint optimization method* with and without the proposed constraints (35) to (37). The effectiveness of the constraints can be demonstrated again from the evolution of the number of the support tokens in optimization. The number does not decrease significantly during the optimization process. Since we use a fixed threshold to select support tokens, if the optimization were to increase the margins for all training samples in any unlimited way, the number of support tokens would have decreased quickly as

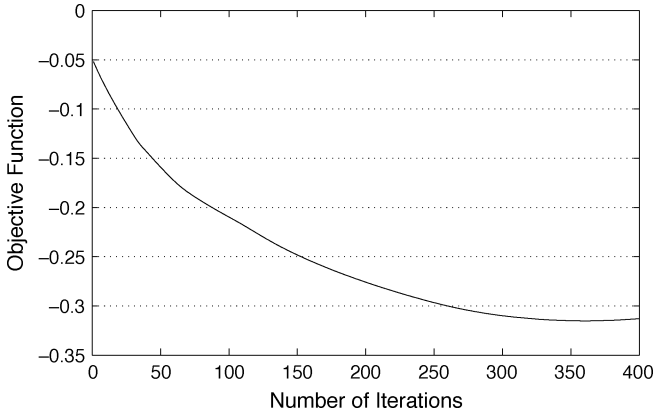


Fig. 2. Objective function versus number of gradient update iterations during the first epoch of constrained joint optimization method for the 1-mix models in the ISOLET E-set task.

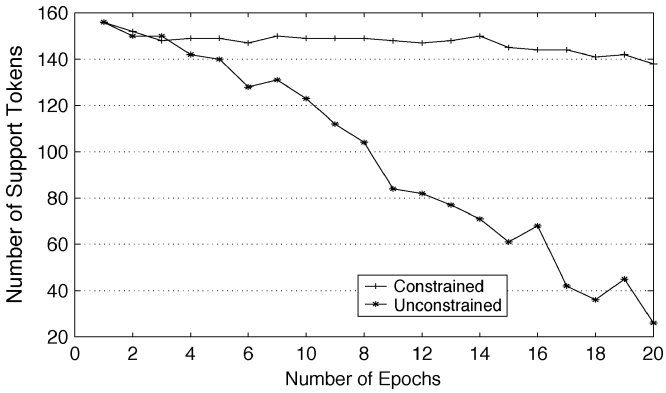


Fig. 3. Number of support tokens is plotted as a function of number of epochs with and without the proposed constraints when the *constrained joint optimization* method is run for the 2-mix models on the E-set task.

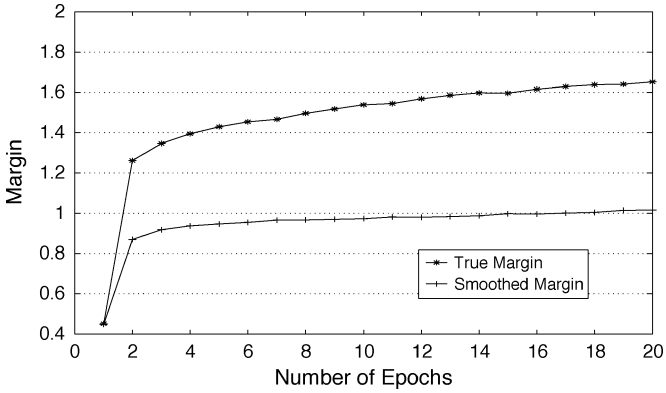


Fig. 4. Real margin and the objective function (i.e., smoothed margin $Q(\Lambda)$) are plotted as a function of number of epochs in the LME training of a 2-mix model on the E-set task.

TABLE I
WORD ACCURACY (IN %) ON THE ISOLET E-SET TEST DATA

	1-mixture	2-mixture	4-mixture
ML	85.56	90.56	**91.48**
MCE	91.48	**94.07**	93.89
LME	92.96	**95.00**	94.44

what we observed if the constraints (35) to (37) were removed from the optimization procedure.

2) Performance of LME Training in E-set: In Table I, we give performance comparison of the best results obtained by using different training criteria to estimate CDHMMs for the E-set recognition. It is clearly demonstrated that the constrained joint optimization method works well in LME training. For example, the LME-trained models with 2-mixture per state achieve the word accuracy of 95.00%, which indicates 15.68% errors reduction over the corresponding MCE-trained models, which get 94.07% in word accuracy. From the experimental results in Table I, we can see that 4-mix models performs slightly worse than 2-mix models. Because we use 16 states for each alphabet model, 4-mix model is slightly over-trained in this small database. At last, if we compare the performance here with our previous results based on a heuristic optimization approach in [22], we can see that both methods perform very similarly in this task. However, the optimization method proposed in this paper is much more solid in theory. It may make difference in other larger scale tasks.

Furthermore, we also plot in Fig. 5 the word accuracy of LME models (with two mixtures) on the test set as a function of number of epochs in the constrained joint optimization method. As shown in Fig. 5, the word recognition accuracy of the 2-mix model on the testing set also increases as the iterative LME training proceeds. After a number of iterations, the LME models achieve 95.00% in word accuracy on the testing set, representing a 15.68% reduction in recognition error over the best MCE models, and a 47% error reduction over the ML models.

B. Continuous Speech Recognition: TIDIGITS Digit Strings

Following [25], the proposed LME training can be easily extended to string-level model for continuous speech recognition. In this case, a couple of string-level competing models are computed for each utterance in training set based on its N-best decoding results. The string-level LME algorithm has been evaluated in a connected digit string recognition task by using the TIDIGITS corpus [21]. This corpus contains utterances from a total of 326 speakers (111 men, 114 women, and 101 children), coming from 21 regions of the United States. The corpus vocabulary is made of the digits of "1" to "9," plus "oh" and "zero," for a total of 11 words. The lengths of the digit strings are from 1 to 7 (except 6). Only adult portion of the corpus is used in our experiments. It contains a total of 225 speakers (111 men and 114 women), 112 of which (55 men and 57 women) are used for training and 113 (56 men and 57 women) for testing. The training set has 8623 digit strings and the test set has 8700 strings.

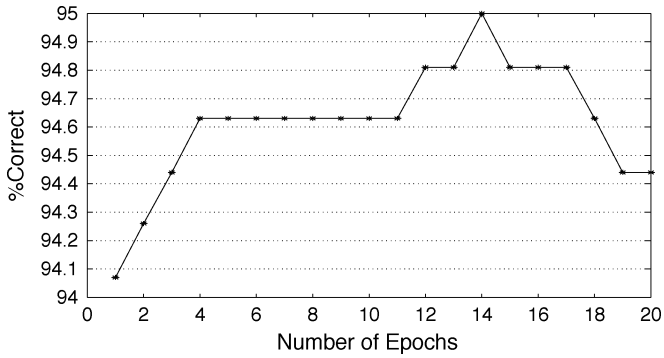


Fig. 5. Word accuracy of LME models on the test set is plotted as a function of number of epochs in the LME training of a 2-mix model on the E-set task.

Our model set has 11 whole-word CDHMMs representing all digits. Each HMM has 12 states and use a simple left-to-right topology without state-skip. The data sampling rate is 16 kHz. Acoustic feature vectors are of standard 39 dimensions (12 MFCCs and the normalized energy, plus their first- and second-order time derivatives). Different number of Gaussian mixture components per state are experimented. The models are tested with unknown-length digit string with maximum seven digits. The ML models are estimated using HTK 3.0. The MCE training uses the best ML model as the seed model. All HMM model parameters (except transition probabilities) are updated during the MCE training process. The MCE algorithm with online update is implemented exactly following [18]. The two parameters of the sigmoid function are experimentally tuned for the best performance on the test set: the slope value $\gamma = 2.0$ and shift value $\theta = 0$ are used in this task. The number of competing strings in the N-Best list is set to five. In the MCE training, the competing strings are combined with smoothing factor $\eta = 10$. For the LME training, we always use the best MCE model as the initial models. As opposed to the MCE training, only Gaussian means are updated during the LME training. Different training parameters are tuned to achieve the best possible model. For example, the threshold ϵ in (4) is set to a values to ensure we have adequate number of tokens in the support token set for every epoch. And the value η in (39) is set dynamically in each epoch to make the largest value of $\eta \cdot d_{ij}(X_i)$ around -10.0. The values of τ_1 and τ_2 in (40) are set to 100. The step size ϵ in the gradient descent search is set between 0.02 and 0.04. G_{ij} in (23) is set to be 0.01. In the TIDIGITS task, recognition accuracy on the training set is less than 100% even after the MCE and LME training. The recognition results (in string accuracy) on the training set are given in Table II for various model complexity. It is clear that the LME training can significantly reduce recognition errors in training data set, ranging from 55% to 84% relative error reduction. During the LME training, from one epoch to next, some of the training data with negative margins may become positive. In this case, they will be included in the support token set in the next epoch. In the TIDIGITS task, we limit to run at most 20 iterations during each epoch due to the efficiency reason.

In Table III, we give string recognition accuracy in the test set of TIDIGITS for the best models obtained by different training

TABLE II
STRING ACCURACY (%) FOR DIFFERENT MODELS IN THE TIDIGIT TRAINING SET. THE NUMBERS INSIDE PARENTHESES REPRESENT THE RELATIVE STRING ERROR REDUCTION OF LME OVER MCE

	ML	MCE	LME
1-mix	89.20	94.61	98.30(68%)
2-mix	95.99	96.82	99.49(84%)
4-mix	97.89	98.39	99.36(60%)
8-mix	98.93	99.04	99.61(59%)
16-mix	99.22	99.26	99.76(67%)
32-mix	99.44	99.56	99.80(55%)

TABLE III
STRING ACCURACY (%) FOR DIFFERENT MODELS IN THE TIDIGITS TEST SET. THE NUMBERS INSIDE PARENTHESES REPRESENT THE RELATIVE STRING ERROR REDUCTION OF LME OVER MCE

	ML	MCE	LME
1-mix	87.39	93.28	96.23(44%)
2-mix	94.74	96.06	98.30(57%)
4-mix	96.52	97.77	98.76(44%)
8-mix	98.06	98.59	99.13(38%)
16-mix	98.28	98.89	99.18(26%)
32-mix	98.66	99.10	**99.34** (27%)

criteria. The results are listed for various model sizes we have investigated. The results clearly show the LME training method considerably reduces recognition error rates on top of the MCE training across all different model sizes. As the model size gets bigger and error rate gets smaller, the advantages of using LME decreases, but still remains significant. For small model sizes (such as 1-mix, 2-mix, 4-mix, 8-mix), the LME training method typically yields over 40% relative error reduction on top of the MCE training. For large model sizes (such as 16-mix and 32-mix), the LME method still gives around 26%–27% error reduction from the MCE discriminative training.

As the final remark, in this paper, the proposed LME training method achieves string error rate 0.66% and word error rate 0.22% in the TIDIGITS task. To our best knowledge, this is the best result reported so far in this task.[4]

VI. CONCLUSION

In this paper, we have studied estimating Gaussian mixture CDHMMs based on the principle of maximizing the minimum multiclass separation margin. We have formulated the problem as a minimax optimization under some nonlinear constraints. At last, we proposed to solve this constrained minimax optimization problem by using a penalized gradient descent search algorithm. The method has been successfully applied to two standard speech recognition tasks, namely the E-set recognition task with the ISOLET database and the connected-digit string recognition with the TIDIGITS database. In both cases, recognition error rates have been significantly reduced with the proposed large margin training approach. This paper shows the proposed framework of the so-called large margin HMMs is superior to other early efforts to combine SVM with HMM in speech recognition. More importantly, the new framework looks very promising to be capable of solving other larger scale speech recognition tasks as well. Some extensive research works are under way to extend the large margin training method to subword-based large vocabulary continuous speech recognition tasks and to investigate how to handle misrecognition utterances in the training set.

[4]In [31], 0.89% string error rate was reported with the MMI training. In [18], under a similar setting of context-independent digit model as our work, 0.95% string error rate was reported with the MCE, and 0.72% string error rate and 0.24% word error rate was reported with the complicated context-dependent head-body-tail models after the MCE training.

REFERENCES

[1] Y. Altun, I. Tsochantaridis, and T. Hofmann, "Hidden Markov support vector machines," in *Proc. 20th Int. Conf. Mach. Learning (ICML)*, Washington, DC, 2003.

[2] J. Arenas-Garcia and F. Perez-Cruz, "Multi-class support vector machines: a new approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2003, pp. II-781–II-784.

[3] 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," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Tokyo, Japan, 1986, pp. 49–52.

[4] I. Bazzi and D. Katabi, "Using support vector machines for spoken digit recognition," in *Proc. Int. Conf. Spoken Lang. Process.*, Beijing, China, 2000, pp. 433–436.

[5] C. J. C. Burges, "A tutorial on support vector machine for pattern recognition," *Data Mining and Knowledge Discovery*, no. 2, pp. 121–167, 1998.

[6] P. Clarkson and P. J. Moreno, "On the use of support vector machine for phonetic classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 15–19, 1999, pp. 585–588.

[7] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learning Res.*, vol. 2, pp. 265–292, 2001.

[8] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, D. Mrva, P. C. Woodland, and K. Yu, "Training LVCSR systems on thousands of hours of data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2005, pp. I-209–I-212.

[9] S. Fine, G. Saon, and R. A. Gopinath, "Digit recognition in noisy environments via a sequential GMM/SVM system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2002, pp. I-49–I-52.

[10] A. Ganapathisraju, J. Hamaker, and J. Picone, "Support vector machines for speech recognition," in *Proc. Int. Conf. Spoken Language Process.*, 1998.

[11] ——, "Hybrid SVM/HMM architecture for speech recognition," in *Proc. Int. Conf. Spoken Language Process.*, Beijing, China, Oct. 2000, pp. 504–507.

[12] S. E. Golowich and D. X. Sun, "A support vector/hidden Markov model approach to phoneme recognition," in *ASA Proc. Statist. Comput. Section*, 1998, pp. 125–130.

[13] X. He and W. Chou, "Minimum classification error linear regression for acoustic model adaptation of CDHMMs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2003, pp. I-556–I-559.

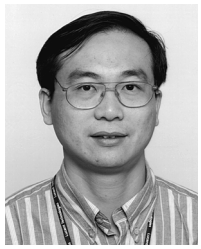
[14] T. Jaakkola and D. Hausser, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems 11*, S. A. Solla and D. A. Cohn, Eds. Cambridge, MA: MIT Press, pp. 487–493.

[15] H. Jiang, K. Hirose, and Q. Huo, "Robust speech recognition based on Bayesian prediction approach," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 4, pp. 426–440, Jul. 1999.

[16] H. Jiang, "Discriminative training for large margin HMMs," Dept. Comput. Sci. Eng., York Univ., Tech. Rep. CS-2004-01, Mar. 2004.

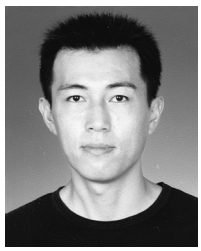
[17] H. Jiang, F. Soong, and C.-H. Lee, "A dynamic in-search data selection method with its applications to acoustic modeling and utterance verification," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 945–955, Sep. 2005.

[18] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.
[19] S. Katagiri, B.-H. Juang, and C.-H. Lee, "Pattern recognition using a generalized probabilistic descent method," *Proc. IEEE*, vol. 86, no. 11, pp. 2345–2373, Nov. 1998.
[20] M. I. Layton and M. J. F. Gales, "Maximum margin training of generative kernels," Cambridge Univ., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR.484, Jun. 2004.
[21] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1984, pp. 328–331.
[22] X. Li, H. Jiang, and C. Liu, "Large margin HMMs for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, Mar. 2005, pp. V513–V516.
[23] X. Li and H. Jiang, "A constrained joint optimization method for large margin HMM estimation," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, 2005, pp. 331–336.
[24] C. Liu, H. Jiang, and X. Li, "Discriminative training of CDHMMs for maximum relative separation margin," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, Mar. 2005, pp. V101–V104.
[25] C. Liu, H. Jiang, and L. Rigazio, "Maximum relative margin estimation of HMMs based on N-best string models for continuous speech recognition," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, 2005, pp. 418–423.
[26] W. Macherey, L. Haferkamp, R. Schluter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Proc. Eur. Conf. Speech Commun. Technol.*, Sep. 2005, pp. 2133–2136.
[27] E. McDermott and S. Katagiri, "A derivation of minimum classification error from the theoretical classification risk using Parzen estimation," *Comput. Speech Lang.*, vol. 18, pp. 107–122, 2004.
[28] E. McDermott and T. J. Hazen, "Minimum classification error training of landmark models for real-time continuous speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2004, pp. I-937–I-940.
[29] P. J. Moreno and P. P. Ho, "A new SVM approach to speaker identification and verification using probabilistic distance kernels," in *Proc. Eur. Conf. Speech Commun. Technol.*, Geneva, Switzerland, Sep. 2003, pp. 2965–2968.
[30] P. Niyogi, C. Burges, and P. Ramesh, "Distinctive feature detection using support vector machines," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1999, pp. 425–428.
[31] Y. Normandin, R. Cardin, and R. Demori, "High-performance connected digit recognition using maximum mutual information estimation," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 299–311, Apr. 1994.
[32] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Orlando, FL, 2002, pp. 105–108.
[33] J. Le Roux and E. McDermott, "Optimization methods for discriminative training," in *Proc. Eur. Conf. Speech Commun. Technol.*, Sep. 2005, pp. 3341–3344.
[34] R. Schluter and H. Ney, "Model-based MCE bound to the true Bayes' error," *IEEE Signal Process. Lett.*, vol. 8, no. 5, pp. 131–133, May 2001.
[35] B. Scholkopf and A. J. Smola, *Learning With Kernels: Support Vector Machine, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
[36] H. Shimodaira, K. Noma, M. Nakai, and S. Sagayama, "Support vector machine with dynamic time-alignment kernel for speech recognition," in *Proc. Eur. Conf. Speech Commun. Technol.*, Aalborg, Denmark, 2001, pp. 1841–1844.
[37] N. Smith and M. Niranjan, "Data-dependent kernels in SVM classification of speech signals," in *Proc. Int. Conf. Spoken Language Process.*, Beijing, China, Oct. 2000, pp. 297–300.
[38] N. Smith and M. Gales, "Speech recognition using SVMs," *Advances in Neural Inf. Process. Syst.*, pp. 1197–1204, 2001.
[39] A. J. Smola, P. Bartlett, B. Scholkopf, and D. Schuurmans, Eds., *Advances in Large Margin Classifiers*. Cambridge, MA: MIT Press, 2000.
[40] J. Stadermann and G. Rigoll, "A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP)*, Jeju Island, Korea, 2004, pp. 661–664.
[41] V. Wan and W. M. Campbell, "Support vector machines for speaker verification and identification," in *Proc. Neural Netw. Signal Process. X*, 2000, pp. 775–784.
[42] V. Wan and S. Renals, "Evaluation of kernel methods for speaker verification and identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2002, pp. I-669–I-672.
[43] J. Weston and C. Watkins, "Support vector machines for multiclass pattern recognition," in *Proc. Eur. Symp. Artif. Neural Netw.*, 1999.
[44] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[45] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–47, Jan. 2002.



Hui Jiang (M'00) received B.Eng. and M.Eng. degrees from University of Science and Technology of China (USTC), Hefei, and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in September 1998, all in electrical engineering.

From October 1998 to April 1999, he was a Researcher with the University of Tokyo. From April 1999 to June 2000, he was with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as a Postdoctoral Fellow. From 2000 to 2002, he worked in the Dialogue Systems Research, Multimedia Communication Research Laboratory, Bell Labs, Lucent Technologies, Inc., Murray Hill, NJ. Since Fall 2002, he has been with the Department of Computer Science and Engineering, York University, Toronto, ON, Canada, as an Assistant Professor. His current research interests include speech and audio processing, machine learning, statistical data modeling, and bioinformatics, especially discriminative training, robustness, noise reduction, utterance verification, and confidence measures.



Xinwei Li received the B.S. degree in electronics from Beijing University, Beijing, China, and the M.S. degree in computer science from York University, Toronto, ON, Canada.

He is a Speech Scientist with Nuance Communications, Burlington, MA. His major research interest focuses on automatic speech recognition, especially discriminative training.

Chaojun Liu (M'05) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, in 1994, the M.S. degree in acoustics from the Institute of Acoustics, Chinese Academy of Science, Beijing, China, in 1997, and the Ph.D. degree in computer science from the OGI School of Science and Engineering, Oregon Health and Science University (formerly Oregon Graduate Institute), Portland, in 2002.

He joined the Fonix Corporation, Salt Lake City, UT, in 2002 as a Senior Scientist and became Director for Core Technology in 2003. In 2004, he worked as a Postdoctoral Fellow at York University, Toronto, ON, Canada and later as a Senior Scientist in Conversay Corporation, Seattle, WA. Since 2005, he has been with the Panasonic Digital Networking Laboratory (now Panasonic San Jose Laboratory), Panasonic R&D Company of America, San Jose, CA, as a Senior Research Engineer, working on various aspects of speech recognition. His current research interests include large-vocabulary speech recognition, discriminative training algorithms, multimedia indexing, and retrieval.