Solving Large-Margin Hidden Markov Model Estimation via Semidefinite Programming

Xinwei Li and Hui Jiang, *Member, IEEE*

***Abstract*—In this paper, we propose to use a new optimization method, i.e., *semidefinite programming (SDP)*, to solve the large-margin estimation (LME) problem of continuous-density hidden Markov model (CDHMM) in speech recognition. First, we introduce a new constraint for LME to guarantee the boundedness of the margin of CDHMM. Second, we show that the LME problem subject to this new constraint can be formulated as an SDP problem under some relaxation conditions. Therefore, it can be solved using many efficient optimization algorithms specially designed for SDP. The new LME/SDP method has been evaluated on a speaker independent E-set speech recognition task using the ISOLET database and a connected digit string recognition task using the TIDIGITS database. Experimental results clearly demonstrate that large-margin estimation via semidefinite programing (LME/SDP) can significantly reduce word error rate (WER) over other existing CDHMM training methods, such as MLE and MCE. It has also been shown that the new SDP-based method largely outperforms the previously proposed LME optimization methods using gradient descent search.**

***Index Terms*—Continuous-density hidden Markov models (CDHMMs), convex optimization, large-margin classifers, large-margin hidden Markov models, semidefinite programming (SDP).**

I. Introduction

RECENTLY, it has been shown that discriminative training techniques, such as maximum mutual information (MMI) and minimum classification error (MCE), can significantly improve speech recognition performance over the conventional maximum-likelihood (ML) estimation. More recently, we have proposed large-margin HMMs for speech recognition [13], [15], [16], [20], where continuous-density hidden Markov models (CDHMMs) are estimated based on the principle of maximizing the minimum margin. From the theoretical results in machine learning [27], a large-margin classifier implies good generalization power and generally yields much lower classification errors in unseen test data. Readers can refer to [13] for a detailed survey about previous relevant works to apply machine learning methods to automatic speech recognition (ASR). As in [15] and [16], large-margin estimation (LME) of CDHMM turns out to be a constrained minimax optimization problem. As opposed to hidden Markov support machine in [2] and Max-Margin Markov Network (M^3-net) in [24], LME of CDHMM cannot be formulated as a relatively simple quadratic programming (QP) problem due to the involved quadratic constraints. In the past few years, several optimization methods have been proposed to solve LME, such as *iterative localized optimization* in [15] and *constrained joint optimization method* in [13] and [16]. Although the constrained minimax problem in *constrained joint optimization method* can be converted into an unconstrained minimization problem by casting the constraints as penalty terms in objective function as in [13] and [16], it remains as a nonlinear and nonconvex optimization problem. For general nonconvex optimization problems, searching global optimum is very difficult, especially in a high-dimensionality space. The steepest gradient descent method used in [13] and [16] can only lead to a locally optimal solution which highly depends on the initial models used for the optimization. And the gradient descent search can be easily trapped into a shallow local optimum when the objective function is jagged and complicated. Moreover, the gradient descent search is hard to control in practice since there are a number of sensitive parameters we need to manually tune for various experimental settings, such as the penalty coefficients and step size and so on. Besides, another closely related work on large-margin estimation of HMMs has been recently reported in [22].

In this paper, we propose to use a better optimization method for LME of CDHMM in speech recognition. First of all, we introduce a new constraint to bound the margin of CDHMM in LME. Under this new constraint, the LME problem can be easily converted into a *semidefinite programming (SDP)* problem under some relaxation conditions. In this way, we are able to take advantage of many efficient SDP algorithms [6], [7], [9], [17], [23], [26] to solve the LME of CDHMM for speech recognition. SDP is an extension of linear programming (LP), and most interior-point methods [17] for LP can be generalized to solve SDP problems. As in LP, these algorithms possess polynomial worst case complexity under certain computation models, but they usually perform very well in practice in terms of efficiency. More importantly, these algorithms can lead to the globally optimal solution since SDP is a well-defined convex optimization problem. In this paper, large-margin CDHMMs estimated with the proposed SDP optimization method are evaluated on a speaker independent E-set speech recognition task using the ISOLET database and a connected digit string recognition task using the TIDIGITS database. Experimental results show that the newly proposed

Manuscript received November 8, 2006; revised June 24, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mary P. Harper.

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

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

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2007.905151

SDP method is very effective in terms of recognition accuracy and optimization efficiency. The SDP-based optimization yields significantly better performance than the previously proposed gradient descent-based methods in [13], [15], and [16]. With the SDP-based optimization method, the LME models achieve 0.53% in string error rate and 0.18% in word error rate (WER) on the TIDIGITS task, which is one of the best results ever reported on this task.

The remainder of this paper is organized as follows. First of all, in Section II, we will briefly review the large-margin estimation criterion for HMM. Next, in Section III, a new constraint is introduced for large-margin estimation to bound margin of CDHMM. Then, we will present how to convert the constrained minimax problem in LME into an SDP problem under some relaxation conditions in Section IV. Experimental results on the ISOLET and TIDIGITS databases will be reported and discussed in Section V. At last, we will conclude the paper with our findings in Section VI.

II. Large-Margin HMMs For ASR

As in [13], the separation margin for a speech utterance X_i in a multiclass classifier can be defined as

$$\begin{aligned} d(X_i) &= \mathcal{F}(X_i;\lambda_{W_i}) - \max_{j\in\Omega j\neq W_i} \mathcal{F}(X_i;\lambda_j) \\ &= \min_{j\in\Omega j\neq W_i} [\mathcal{F}(X_i;\lambda_{W_i}) - \mathcal{F}(X_i;\lambda_j)] \end{aligned} \quad (1)$$

where Ω denotes the set of all possible words or word sequences, λ_W or λ_j denotes the concatenated HMM representing a word or word sequence in Ω, W_i represents the true transcription of X_i, and $\mathcal{F}(X;\lambda_W)$ is called discriminant function. Usually, the discriminant function is calculated in the logarithm scale: $\mathcal{F}(X;\lambda_W) = \log[p(W)\cdot p(X|\lambda_W)]$. In this paper, we are only interested in estimating HMMs λ_W and assume $p(W)$ is fixed. 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 models Λ.

Given a set of training data $\mathcal{D} = \{X_1, X_2, \ldots, X_N\}$, we usually know the true transcriptions for all utterances in $\mathcal{D}$, denoted as $\mathcal{L} = \{W_1, W_2, \ldots, W_N\}$. The *support vector set* $\mathcal{S}$ is defined as

$$\mathcal{S} = \{X_i | X_i \in \mathcal{D} \text{ and } 0 \leq d(X_i) \leq \gamma\} \quad (2)$$

where $\gamma > 0$ is a preset positive threshold. All utterances in $\mathcal{S}$ are relatively close to the classification boundary even though all of them locate in right decision region.

The large-margin principle leads to estimating 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

$$\begin{aligned} \Lambda^* &= \arg\max_{\Lambda} \min_{X_i\in\mathcal{S}} d(X_i) \\ &= \arg\min_{\Lambda} \max_{X_i\in\mathcal{S} j\in\Omega j\neq W_i} [\mathcal{F}(X_i;\lambda_j) - \mathcal{F}(X_i;\lambda_{W_i})]. \end{aligned} \quad (3)$$

Note that the support token set $\mathcal{S}$ is selected and used in LME because the other training data with larger margin are usually inactive in optimization towards maximizing the minimum margin.

III. New Constraint For Large-Margin Estimation

As shown in [13], [15], [16], and [20], the margin as defined in (1) is actually unbounded for CDHMM. In other words, we can adjust CDHMM parameters in a way to increase the margin unlimitedly. In [13] and [16], we introduced some theoretically sounded constraints to bound the margin to ensure the existence of the optimal point in the large-margin estimation of (3). However, it is difficult to formulate the constrained minimax optimization in [13] and [16] into an SDP problem. In this section, we introduce a new locality constraint which prevents model parameters from deviating too far from their initial values. More importantly, LME of CDHMM under this constraint can be easily converted into an SDP problem.

First of all, suppose each speech unit, e.g., a phoneme, is modeled by an N-state CDHMM with parameter vector $\lambda = (\pi, A, \theta)$, where π is the initial state distribution, $A = \{a_{ij}|1 \leq i,j \leq N\}$ is transition matrix, and θ is parameter vector composed of mixture parameters $\theta_i = \{\omega_{ik}, \boldsymbol{\mu}_{ik}, \Sigma_{ik}\}_{k=1,2,\ldots,K}$ for each state i, where K denotes number of Gaussian mixtures in each state. The state observation probablity density function (pdf) is assumed to be a mixture of multivariate Gaussian distribution

$$\begin{aligned} p(\mathbf{x}|\theta_i) &= \sum_{k=1}^{K} \omega_{ik}\cdot\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{ik},\Sigma_{ik}) \\ &= \sum_{k=1}^{K} \omega_{ik}\cdot(2\pi)^{-D/2}|\Sigma_{ik}|^{-1/2} \\ &\quad\times\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{ik})^T\Sigma_{ik}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{ik})\right] \end{aligned} \quad (4)$$

where D denotes dimension of feature vector, and mixture weights ω_{ik}'s satisfy the constraint $\sum_{k=1}^{K}\omega_{ik} = 1$. In many cases, we prefer to use diagonal covariance matrices. Thus, the above state observation pdf can be simplified as

$$p(\mathbf{x}|\theta_i) = \sum_{k=1}^{K}\omega_{ik}\cdot\prod_{d=1}^{D}\sqrt{\frac{1}{2\pi\sigma_{ikd}^2}}e^{(-(x_d-\mu_{ikd})^2/2\sigma_{ikd}^2)}. \quad (5)$$

Given any speech utterance $X_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{iT}\}$, if we adopt the Viterbi approximation as in [15] and [16], we know that the discriminant function based on model λ_j, i.e., $\mathcal{F}(X_i;\lambda_j)$, can be expressed as

$$\begin{aligned} \mathcal{F}(X_i;\lambda_j) \approx{}& \log p(W_j) + \log\pi_{s_1^*} \\ &+ \sum_{t=2}^{T}\log a_{s_{t-1}^* s_t^*} + \sum_{t=1}^{T}\log\omega_{s_t^* l_t^*} \\ &- \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D}\left[\log\sigma_{s_t^* l_t^* d}^2 + \frac{(x_{itd}-\mu_{s_t^* l_t^* d})^2}{\sigma_{s_t^* l_t^* d}^2}\right] \end{aligned} \quad (6)$$

where we denote the optimal Viterbi path as the best state sequence $\mathbf{s}^* = \{s_1^*, s_2^*, \ldots, s_T^*\}$ and the best mixture component label $\mathbf{l}^* = \{l_1^*, l_2^*, \ldots, l_T^*\}$. Here the best state sequence $\mathbf{s}^*$ is obtained from the Viterbi decoding process. After that, each

l_t^* $(t = 1, 2, \ldots, T)$ is selected as the most probable Gaussian in state s_t^* for each feature vector $\mathbf{x}_t$.

In this paper, for simplicity, we only consider to estimate Gaussian mean vectors of CDHMM based on the large-margin principle while keeping all other parameters constant during large-margin estimation. Therefore, we have

$$\mathcal{F}(X_i; \lambda_j) \approx c''_{ij} - \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{(x_{itd} - \mu_{s_t^* l_t^* d})^2}{\sigma^2_{s_t^* l_t^* d}} \tag{7}$$

where c''_{ij} is a constant which is independent from all Gaussian mean vectors.

Furthermore, we assume there are totally $\mathcal{K}$ distinct Gaussians in the whole CDHMM set Λ. We denote them as $\mathcal{N}(u_k, \Sigma_k)$, where $k \in (1, 2, \ldots, \mathcal{K})$. For notational convenience, the optimal Viterbi path $\mathbf{s}^*$ and $\mathbf{l}^*$ can be equivalently represented as a sequence of Gaussian index, i.e., $\mathbf{j} = \{j_1, j_2, \ldots, j_T\}$, where $j_t \in (1, 2, \ldots, \mathcal{K})$ is the index of each Gaussian mixture along the optimal path $\{\mathbf{s}^*, \mathbf{l}^*\}$. Therefore, we can rewrite the discriminant function in (7) as

$$\mathcal{F}(X_i; \lambda_j) \approx c''_{ij} - \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D}\frac{(x_{itd} - \mu_{j_t d})^2}{\sigma^2_{j_t d}}. \tag{8}$$

Similarly, we assume the Viterbi path for $\mathcal{F}(X_i; \lambda_{W_i})$ as $\mathbf{i} = \{i_1, i_2, \ldots, i_T\}$. Therefore, the decision margin $d_{ij}(X_i)$ can be represented as a standard diagonal quadratic form as follows:

$$\begin{aligned} d_{ij}(X_i) &= \mathcal{F}(X_i; \lambda_{W_i}) - \mathcal{F}(X_i; \lambda_j) \\ &\approx c_{ij} - \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D}\left[\frac{(x_{itd} - \mu_{i_t d})^2}{\sigma^2_{i_t d}} - \frac{(x_{itd} - \mu_{j_t d})^2}{\sigma^2_{j_t d}}\right] \end{aligned} \tag{9}$$

where c_{ij} is a constant independent of all Gaussian means.

Obviously, if every term in summation of (9) is bounded, the decision margin $d_{ij}(X_i)$ is also bounded. It is easy to see that all these items will be bounded if every Gaussian mean $\boldsymbol{\mu}_k$, for any $k \in (1, 2, \ldots, \mathcal{K})$, in the model set is constrained in a limited range. Therefore, we introduce the following spherical constraint for all Gaussian mean vectors:

$$R(\Lambda) = \sum_{k=1}^{\mathcal{K}}\sum_{d=1}^{D}\frac{\left(\mu_{kd} - \mu_{kd}^{(0)}\right)^2}{\sigma^2_{kd}} \le r^2 \tag{10}$$

where r is a pre-set constant, and $\mu_{kd}^{(0)}$ represents the initial values of μ_{kd} in the seed models. The boundedness of the margin $d(X_i)$ can be guaranteed by the following theorem:

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 (1), is bounded for any token X_i in the training set $\mathcal{D}$ as long as the constraint in (10) holds.

It is trivial to prove this theorem since the constraint in (10) defines a closed and compact set (see [18] for the proof). According to theorem III.1, the minimum margin in (3) is a bounded function of model parameter set Λ under the constraint in (10). Thus, we can always search for an appropriate set of model parameters to maximize the minimum margin. Therefore, the minimax optimization problem in (3) becomes solvable under these constraints. Hence, we reformulate the large-margin estimation as the following constrained minimax optimization problem:

1) Problem 1:

$$\Lambda^* = \arg\min_{\Lambda}\max_{X_i \in \mathcal{S} j \in \Omega j \neq W_i}\left[\mathcal{F}(X_i; \lambda_j) - \mathcal{F}(x_i; \lambda_{W_i})\right] \tag{11}$$

subject to

$$R(\Lambda) = \sum_{k=1}^{\mathcal{K}}\sum_{d=1}^{D}\frac{\left(\mu_{kd} - \mu_{kd}^{(0)}\right)^2}{\sigma^2_{kd}} \le r^2 \tag{12}$$

$$\begin{aligned} &\mathcal{F}(X_i; \lambda_j) - \mathcal{F}(X_i; \lambda_{W_i}) \\ &< 0 \end{aligned} \tag{13}$$

for all $X_i \in \mathcal{S}$ and $j \in \Omega$ and $j \neq W_i$. Here, r is a preset constant. $\mu_{kd}^{(0)}$ represents the initial values of μ_{kd} in the seed models.

IV. SDP Formulation of LME

As shown above, large-margin estimation (LME) of CDHMM turns out to be a constrained minimax optimization as shown in *Problem 1*. Obviously, it is a complicated nonlinear optimization problem, where typically no efficient solution exists from the viewpoint of optimization theory. Although this constrained minimax problem can be converted into an unconstrained minimization problem by casting the constraints as the penalty terms in the objective function as shown in [13] and [16], it still remains as a nonlinear nonconvex optimization problem. There is no efficient algorithm available to solve this optimization problem, especially in the case of speech recognition where a large number of model parameters are involved. The gradient descent method used in [13] and [16] can only lead to a locally optimal solution which highly depends on the initial models used for the optimization. And the gradient descent can be easily trapped into a shallow local optimum when the objective function is jagged and complicated. Moreover, the gradient descent search is hard to control in practice since there are a number of sensitive parameters we need to manually tune for various experimental settings, such as the penalty coefficients, the step size, and so on. Since an improper setting of any of these parameters may dramatically deteriorate optimization performance, a large number of experiments are necessary to find the optimal values for these parameters, which makes the LME training time-consuming.

In this section, we will consider to convert the minimax optimization *Problem 1* into an SDP problem under some relaxation conditions. In this way, we are able to take advantage of many efficient SDP algorithms to solve the LME of CDHMM for speech recognition. SDP is more general than linear programming (LP), but SDP is not much harder to solve. As in LP, most SDP algorithms possess polynomial worst case complexity under certain computation models. More importantly, these algorithms can lead to the globally optimal solution since SDP is a well-defined convex optimization problem. We can also

avoid choosing many optimization parameters manually in experiments since most of them can be gracefully handled by the algorithms themselves.

A. Introduction to SDP

In this paper, we generally follow the notations in this section except explicitly stated otherwise. We denote the set of real numbers by $\mathbf{R}$. $\mathbf{R}_+$ denotes the set of nonnegative real numbers. For a natural number n, the symbol $\mathbf{R}^n$ $\left(\mathbf{R}_+^n\right)$ denotes the set of vectors with n components in $\mathbf{R}(\mathbf{R}_+)$. We always denote vectors using bold lowercase letters. Uppercase letters will be used to represent matrices. The vector inequality $\mathbf{x} \geq \mathbf{y}$ means $x_j \geq y_j$ for $j = 1; 2; \ldots; n$. $\mathbf{0}$ represents a vector whose entries are all zeros. And $\mathbf{e}_k$ is a vector with -1 at the kth position and zero everywhere else. The dimensions are decided according to the context in an algebraic expression. A vector is usually considered as a column vector unless otherwise stated. For convenience, we sometimes write a column vector $\mathbf{x}$ as $\mathbf{x} = (x_1; x_2; \ldots; x_n)$ and a row vector as $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. The superscript T denotes transpose operation. The inner product in $\mathbf{R}^n$ is denoted as $\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T\mathbf{y} = \sum_{i=1}^n x_i y_i$ for $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$. For natural numbers m and n, $\mathbf{R}^{m\times n}$ denotes the set of real matrices with m rows and n columns. The identity matrix is denoted by I. The trace of A, denoted as $\mathrm{tr}(A)$, is the sum of the diagonal entries in A. For a vector $\mathbf{x} \in \mathbf{R}^n$, $D_{\mathbf{x}}$ represents a diagonal matrix in $\mathbf{R}^{n\times n}$ whose diagonal entries are the entries of $\mathbf{x}$, i.e., $D_{\mathbf{x}} = \mathrm{diag}(\mathbf{x})$. A matrix $Q \in \mathbf{R}^{n\times n}$ is said to be positive definite, denoted as $Q \succ 0$, if $\mathbf{x}^T Q\mathbf{x} > 0$ for all $\mathbf{x} \neq 0$. And Q is positive semidefinite, denoted as $Q \succeq 0$, if $\mathbf{x}^T Q\mathbf{x} \geq 0$ for all $\mathbf{x}$. $\mathcal{M}^n$ denotes the space of symmetric matrices in $\mathbf{R}^{n\times n}$. The inner product in $\mathcal{M}^n$ is defined as follows: $X \cdot Y = \mathrm{tr}(X^T Y) = \mathrm{tr}(XY^T) = \sum_{i,j} x_{ij} y_{ij}$ for $X, Y \in \mathcal{M}^n$. And $\mathcal{M}_+^n$ denotes the set of positive semidefinite matrices in $\mathcal{M}^n$.

SDP can be viewed as either an extension of LP or a special case of more general conic optimization problem. The standard SDP form is shown in Definition 1.

Definition 1: The standard SDP form is

$$\min_{X_1, X_2, \ldots, X_J} \quad \sum_{j=1}^{J} C_j \cdot X_j$$
$$\text{subject to} \quad \sum_{j=1}^{J} A_{i,j} \cdot X_j = b_i, \quad i = 1, \ldots, I, \;\; X_j \in \mathcal{M}_+^n$$

where $\mathcal{M}_+^n$ denotes the set of symmetric positive semidefinite matrices, $\cdot$ denotes the associated inner product, $A_{i,j}$ and C_j both are symmetric constant matrices, and b_i is constant scalar.

In semidefinite programming (SDP), we minimize a linear function of symmetric matrices in the positive semidefinite matrix cone subject to affine constraints. Similar to the positive orthant cone in linear programming, the positive semidefinite matrix cone is generalized linear and convex. Thus, semidefinite programs are convex optimization problems. And SDP unifies several standard convex optimization problems, such as linear programming, quadratic programming, and convex quadratic minimization with convex quadratic constraints.

B. Transformation of Objective Function

The standard SDP form is a minimization problem, while *Problem 1* from the LME is a minimax optimization problem. So in the first step, we need to transform the minimax optimization problem into a pure minimization problem.

If we introduce a new variable $-\rho(\rho > 0)$ as a common upper bound to represent *max* part in (11) along with the constraints that every item in the minimax optimization must be less than or equal to $-\rho$, we can convert the minimax optimization in (11) into an equivalent minimization problem as follows:

1) Problem 2:

$$\Lambda^* = \arg\min_{\Lambda,\rho} -\rho \tag{14}$$

subject to:

$$\mathcal{F}(X_i; \lambda_j) - \mathcal{F}(X_i; \lambda_{W_i}) \leq -\rho \tag{15}$$

$$R(\Lambda) = \sum_{k=1}^{\mathcal{K}} \sum_{d=1}^{D} \frac{\left(\mu_{kd} - \mu_{kd}^{(0)}\right)^2}{\sigma_{kd}^2} \leq r^2 \tag{16}$$

$$\rho \geq 0. \tag{17}$$

for all $X_i \in \mathcal{S}$ and $j \in \Omega$, $j \neq W_i$.

C. Transformation of Constraint (15)

In the standard SDP form, all variables and coefficients are presented in the form of inner product of matrices. In this part, we will first derive a matrix form for the constraint in (15). We define mean matrix U as a matrix by concatenating all normalized Gaussian mean vectors as its columns as

$$U = (\tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\mu}}_2, \ldots, \tilde{\boldsymbol{\mu}}_{\mathcal{K}}) \tag{18}$$

where each column is a normalized mean vector

$$\tilde{\boldsymbol{\mu}}_k := \left(\frac{\mu_{k1}}{\sigma_{k1}}; \frac{\mu_{k2}}{\sigma_{k2}}; \ldots; \frac{\mu_{kD}}{\sigma_{kD}}\right). \tag{19}$$

Similar to (8), we can rewrite $\mathcal{F}(X_i; \lambda_{W_i})$ as

$$\begin{aligned}\mathcal{F}(X_i; \lambda_{W_i}) &= c_i' - \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D} \frac{(x_{itd} - \mu_{i_t d})^2}{\sigma_{i_t d}^2} \\ &= c_i' - \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D} (\tilde{x}_{itd} - \tilde{\mu}_{i_t d})^2 \\ &= c_i' - \frac{1}{2}\sum_{t=1}^{T} (\tilde{\mathbf{x}}_{it} - \tilde{\boldsymbol{\mu}}_{i_t})^T (\tilde{\mathbf{x}}_{it} - \tilde{\boldsymbol{\mu}}_{i_t})\end{aligned} \tag{20}$$

where $\tilde{\mathbf{x}}_{it}$ denotes a normalized feature vector (in column) for each $\mathbf{x}_{it}$ as

$$\tilde{\mathbf{x}}_{it} := \left(\frac{x_{it1}}{\sigma_{i_t 1}}; \frac{x_{i2}}{\sigma_{i_t 2}}; \ldots; \frac{x_{itD}}{\sigma_{i_t D}}\right). \tag{21}$$

Obviously, we have

$$\begin{aligned}\tilde{\mathbf{x}}_{it}-\tilde{\boldsymbol{\mu}}_{i_t}&=(I^D,U)(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})\\&=\begin{pmatrix} D\left\{\overbrace{\begin{matrix}1&\cdots&0\\ \vdots&\ddots&\vdots\\0&\cdots&1\end{matrix}}^{D}\right. & \overbrace{\begin{matrix}\tilde{\mu}_{11}&\cdots&\tilde{\mu}_{\mathcal{K}1}\\ \vdots&\vdots&\vdots\\ \tilde{\mu}_{1D}&\vdots&\tilde{\mu}_{\mathcal{K}D}\end{matrix}}^{\mathcal{K}}\end{pmatrix}\\&\quad\times\begin{pmatrix}\left.\begin{matrix}\tilde{x}_{t1}\\ \vdots\\ \tilde{x}_{tD}\end{matrix}\right\}D\\ \left.\begin{matrix}0\\ \vdots\\ -1(i_t)\\ \vdots\\ 0\end{matrix}\right\}\mathcal{K}\end{pmatrix}\end{aligned}\tag{22}$$

where U is defined in (18) and $\mathbf{e}_{i_t}$ is a vector defined in Section IV-A.

Therefore

$$\begin{aligned}\mathcal{F}(X_i;\lambda_{W_i})&=c_i'-\frac{1}{2}\sum_{t=1}^{T}(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})^T(I^D,U)^T\\&\quad\times(I^D,U)(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})\\&=c_i'-\frac{1}{2}\sum_{t=1}^{T}(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})^TZ(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})\\&=c_i'-\frac{1}{2}\sum_{t=1}^{T}\mathrm{tr}\left[(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})^TZ\right]\\&=-A_i\cdot Z+c_i'\end{aligned}\tag{23}$$

where A_i and Z are $(D+\mathcal{K})\times(D+\mathcal{K})$ dimensional symmetric matrices defined as

$$A_i=\frac{1}{2}\sum_{t=1}^{T}(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})(\tilde{\mathbf{x}}_{it};\mathbf{e}_{i_t})^T\tag{24}$$

$$Z=(I^D,U)^T(I^D,U)=\begin{pmatrix}I^D&U\\U^T&Y\end{pmatrix}\quad Y=U^TU.\tag{25}$$

Similarly, we can rewrite the discriminant function, $\mathcal{F}(X_i;\lambda_j)$, as

$$\mathcal{F}(X_i;\lambda_j)=-A_j\cdot Z+c_j''$$

where A_j is a $(D+\mathcal{K})\times(D+\mathcal{K})$ dimensional symmetric matrix defined as

$$A_j=\frac{1}{2}\sum_{t=1}^{T}(\tilde{\mathbf{x}}_{it};\mathbf{e}_{j_t})(\tilde{\mathbf{x}}_{it};\mathbf{e}_{j_t})^T.\tag{26}$$

Thus, it is straightforward to convert the constraint in (15) into the following form

$$\mathcal{F}(X_i;\lambda_j)-\mathcal{F}(X_i;\lambda_{W_i})=A_{ij}\cdot Z-c_{ij}\le-\rho\tag{27}$$

where $A_{ij}=A_i-A_j$ and $c_{ij}=c_i'-c_j''$.

D. Transformation of Constraint (16)

In this part, we will convert the constraint (16) into a matrix inequality constraint needed by the standard SDP formulation. Similar as above, $R(\Lambda)$ in (16) can be rewritten as follows:

$$\begin{aligned}R(\Lambda)&=\sum_{k=1}^{\mathcal{K}}\sum_{d=1}^{D}\left(\tilde{\mu}_{kd}-\tilde{\mu}_{kd}^{(0)}\right)^2\\&=\sum_{k=1}^{\mathcal{K}}\left(\tilde{\boldsymbol{\mu}}_k-\tilde{\boldsymbol{\mu}}_k^{(0)}\right)^T\left(\tilde{\boldsymbol{\mu}}_k-\tilde{\boldsymbol{\mu}}_k^{(0)}\right).\end{aligned}\tag{28}$$

Since $-\left(\tilde{\boldsymbol{\mu}}_k-\tilde{\boldsymbol{\mu}}_k^{(0)}\right)=(I^D;U)\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)$, we can similarly represent this constraint as

$$\begin{aligned}R(\Lambda)&=\sum_{k=1}^{\mathcal{K}}\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)^T(I^D;U)^T(I^D;U)\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)\\&=\sum_{k=1}^{\mathcal{K}}\mathrm{tr}\left[\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)^TZ\right]\\&=Q\cdot Z\le r^2\end{aligned}\tag{29}$$

where Q is a $(D+\mathcal{K})\times(D+\mathcal{K})$-dimensional symmetric matrix defined as

$$Q=\sum_{k=1}^{\mathcal{K}}\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)\left(\tilde{\boldsymbol{\mu}}_k^{(0)};\mathbf{e}_k\right)^T\tag{30}$$

and $\tilde{\boldsymbol{\mu}}_k^{(0)}$ is defined as in (19).

E. SDP Formulation of LME

Substituting (27) and (29) in *Problem 2*, we can formulate the LME problem as

1) Problem 3:

$$\min_{z,\rho}-\rho\tag{31}$$

subject to

$$A_{ij}\cdot Z+\rho\le c_{ij}\tag{32}$$

$$Q\cdot Z\le r^2\tag{33}$$

$$Z=\begin{pmatrix}I^D&U\\U^T&Y\end{pmatrix},\quad Y=U^TU\ \ \rho\ge0.\tag{34}$$

for all $X_i\in\mathcal{S}$ and $j\in\Omega$, $j\ne W_i$.

In order to change the inequalities (32) and (33) to equalities, we introduce a slack variable $s_{(i,j)}$ $(s_{(i,j)}>0)$ for each of the constraints in (32), and a slack variable ω $(\omega>0)$ for constraint (33). We first arrange all the $s_{(i,j)}$ in some order and put them together as a vector $\mathbf{s}=(s_{(i,j)})$ where each (i,j) corresponds to a position in the vector. Then, we can organize all these slack variables and ρ into a semidefinite diagonal matrix

$$S=\mathrm{diag}(\rho;\omega;\mathbf{s}).\tag{35}$$

Next, we denote a few constant diagonal matrices as follows:

$$B_{ij} = \mathrm{diag}(1; 0; -\mathbf{e}_{(i,j)}) \tag{36}$$

$$G = \mathrm{diag}(-1; \mathbf{0}) \tag{37}$$

$$H = \mathrm{diag}(0; 1; \mathbf{0}) \tag{38}$$

where $\mathbf{e}_{(i,j)}$ is a vector with -1 at the corresponding position of $s_{(i,j)}$ in $\mathbf{s}$ and 0 anywhere else.

Then, we have the following:

$$s_{(i,j)} + \rho = B_{ij} \cdot S \tag{39}$$

$$-\rho = G \cdot S \tag{40}$$

$$\omega = H \cdot S. \tag{41}$$

Therefore, the large-margin HMM problem is transformed to the following form:

2) Problem 4:

$$\min_{z,S}[0 \cdot Z + G \cdot S] \tag{42}$$

subject to

$$A_{ij} \cdot Z + B_{ij} \cdot S = c_{ij} \tag{43}$$

$$Q \cdot Z + H \cdot S = r^2 \tag{44}$$

$$Z = \begin{pmatrix} I^D & U \\ U^T & Y \end{pmatrix}, \quad Y = U^T U$$

$$S \succeq 0 \tag{45}$$

for all $X_i \in \mathcal{S}$ and $j \in \Omega\ j \neq W_i$.

Obviously, the minimization *Problem 4* is equivalent to the original minimax optimization *Problem 1* from LME. However, since the constraint $Y = U^T U$ is not convex, it is a nonconvex optimization problem. As shown in [5], the following statement always holds for matrices:

$$Y - U^T U \succeq 0 \Leftrightarrow Z = \begin{pmatrix} I^D & U \\ U^T & Y \end{pmatrix} \succeq 0. \tag{46}$$

Therefore, following [5], if we relax the constraint $Y = U^T U$ to $Y - U^T U \succeq 0$, we are able to make Z a positive semidefinite matrix. During the optimization, the top left corner of Z must be an identity matrix, i.e., $Z_{1:D,1:D} = I^D$, which can be easily transformed into a group of linear constraints as required in the standard SDP form

$$\left[(\mathbf{e}_k + \mathbf{e}_l)(\mathbf{e}_k + \mathbf{e}_l)^T\right] \cdot Z = 2 + 2\delta(k - l) \text{ for } 1 \leq k \leq D,$$
$$k \leq l \leq D. \tag{47}$$

If $k = l$ (for $1 \leq k, l \leq D$)

$$\left[(\mathbf{e}_k + \mathbf{e}_l)(\mathbf{e}_k + \mathbf{e}_l)^T\right] \cdot Z = 4z_{kk} = 4 \tag{48}$$

otherwise

$$\begin{aligned}\left[(\mathbf{e}_k + \mathbf{e}_l)(\mathbf{e}_k + \mathbf{e}_l)^T\right] \cdot Z &= z_{kk} + z_{ll} + z_{kl} + z_{lk} = z_{kk} + z_{ll} + 2z_{kl} \\ &= 2.\end{aligned} \tag{49}$$

Since Z is a symmetric matrix, $z_{kl} = z_{lk}$. Obviously, the solution to this set of constraints is $z_{kk} = 1$ and $z_{kl} = z_{lk} = 0$ for all $1 \leq k \leq D$, $k < l \leq D$.

Finally, under the relaxation in (46), *Problem 4* is converted into a standard SDP problem as

3) Problem 5:

$$\min_{z,S}[0 \cdot Z + G \cdot S] \tag{50}$$

subject to

$$A_{ij} \cdot Z + B_{ij} \cdot S = c_{ij} \tag{51}$$

$$Q \cdot Z + H \cdot S = r^2 \tag{52}$$

$$Z_{1:D,1:D} = I^D. \tag{53}$$

$$S \succeq 0 \quad Z \succeq 0. \tag{54}$$

for all $X_i \in \mathcal{S}$ and $j \in \Omega\ j \neq W_i$.

Problem 5 is a standard SDP problem, which can be solved efficiently by many SDP algorithms. In *Problem 5*, the optimization is carried out w.r.t. Z (which is constructed from all HMM Gaussian means) and S, A_{ij}, c_{ij}, and Q are constants calculated from training data and initial models, and r is a preset parameter, and B_{ij}, G, and H are defined in (36)–(38).

F. Analysis of SDP Relaxation

Apparently, due to the relaxation in (46), this SDP problem is just an approximation to the original LME problem. Now, we investigate how the SDP relaxation in (46) affects the results of large-margin HMMs estimation. Let us define

$$V = Y - U^T U \succeq 0. \tag{55}$$

The actual margin which is maximized in this SDP *Problem 5* after the relaxation can be calculated as

$$\begin{aligned}-d^*_{ij}(X_i) &= A_{ij} \cdot Z - c_{ij} \\ &= A_{ij} \cdot \begin{pmatrix} I^D & U \\ U^T & U^T U \end{pmatrix} - c_{ij} + A_{ij} \cdot \begin{pmatrix} 0 & 0 \\ 0 & V \end{pmatrix} \\ &= -d_{ij}(X_i) + A_{ij} \cdot \begin{pmatrix} 0 & 0 \\ 0 & V \end{pmatrix}.\end{aligned} \tag{56}$$

Thus, the actual margin that we try to maximize in the SDP *Problem 5* is the original margin $d_{ij}(X_i)$ defined in (9) combined with another item

$$A_{ij} \cdot \begin{pmatrix} 0 & 0 \\ 0 & V \end{pmatrix} = \frac{1}{2}\sum_{t=1}^{T} v_{i_t i_t} - \frac{1}{2}\sum_{t=1}^{T} v_{j_t j_t} \tag{57}$$

where $v_{i_t i_t}$ and $v_{j_t j_t}$ are diagonal elements of V at positions (i_t, i_t) and (j_t, j_t).

Substituting (57), (27), and (20) to (56), we derive the actual margin to be maximized in the SDP problem as follows:

$$\begin{aligned} d_{ij}^*(X_i) &= -\frac{1}{2}\sum_{t=1}^{T}\left[(\tilde{\mathbf{x}}_{it}-\tilde{\boldsymbol{\mu}}_{i_t})^T(\tilde{\mathbf{x}}_{it}-\tilde{\boldsymbol{\mu}}_{i_t})+v_{i_ti_t}\right] \\ &\quad +\frac{1}{2}\sum_{t=1}^{T}\left[(\tilde{\mathbf{x}}_{it}-\tilde{\boldsymbol{\mu}}_{j_t})^T(\tilde{\mathbf{x}}_{it}-\tilde{\boldsymbol{\mu}}_{j_t})+v_{j_tj_t}\right]+c_{ij} \\ &= -\frac{1}{2}\sum_{t=1}^{T}(\bar{\mathbf{x}}_{it}-\bar{\boldsymbol{\mu}}_{i_t})^T(\bar{\mathbf{x}}_{it}-\bar{\boldsymbol{\mu}}_{i_t}) \\ &\quad +\frac{1}{2}\sum_{t=1}^{T}(\bar{\mathbf{x}}_{it}-\bar{\boldsymbol{\mu}}_{j_t})^T(\bar{\mathbf{x}}_{it}-\bar{\boldsymbol{\mu}}_{j_t})+c_{ij} \end{aligned} \tag{58}$$

where

$$\bar{\mathbf{x}}_t := (\tilde{\mathbf{x}}_t; 0) \tag{59}$$

$$\bar{\boldsymbol{\mu}}_{i_t} := (\tilde{\boldsymbol{\mu}}_{i_t}; \sqrt{v_{i_ti_t}}) \tag{60}$$

$$\bar{\boldsymbol{\mu}}_{j_t} := (\tilde{\boldsymbol{\mu}}_{j_t}; \sqrt{v_{j_tj_t}}). \tag{61}$$

Comparing (58) with (27) and (20), we can see that this SDP problem actually augments each D-dimension speech feature vector $\mathbf{x}_{it}$ to a $(D+1)$-dimensional vector $\bar{\mathbf{x}}_{it}$ and augments each Gaussian mean vector $\boldsymbol{\mu}_{i_t}$ with a diagonal element of matrix V. And then it tries to maximize a variant margin, $d_{ij}^*(X_i)$ in (58), in this augmented $(D+1)$-dimension space.

G. Summary of the Training Process

As a remark, the 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 (2) and obtain the optimal Viterbi paths for each support token according to its recognition result. Then, the relaxed SDP optimization, i.e., *Problem 5*, is conducted with respect to Z and S. At last, all CDHMM Gaussian mean vectors are updated based on the optimization solution Z^* obtained by the SDP solver. If not convergent, the next epoch starts from recognizing all training data again.

When we update CDHMM Gaussian means with the solution matrix Z^*, we have two methods. 1) *Projection Method*: All Gaussian mean vectors are simply updated with the top-right portion, i.e., U in (25), of the solution matrix Z^* and just discard the remaining part of matrix Z^*, e.g., the right-bottom part Y. This step can be viewed as a projection from a high-dimension space to a low-dimension space. The SDP algorithms guarantee to find the globally optimal solution for the margin in the augmented higher dimension space as in (56), but not the global optimum in the original space after the projection. 2) *Augmentation method*: In this method, we use the whole matrix Z^* (including Y) to update models. We first calculate matrix V based on Z^* as in (55). Next, we update Gaussian mean vectors with the right-top part U in Z^*; then, we augment each Gaussian mean vector with its corresponding diagonal element of matrix V to create a $(D+1)$-dimension CDHMM as in (60) or (61), which will be used for recognition directly in a higher-dimension space. When we recognize with this augmented model, we also need to augment a zero to each feature vector to derive a $(D+1)$-dimension vector $\bar{\mathbf{x}}_{it}$ in (59).

V. Experiments

In this paper, we evaluate the SDP-based LME training method on two speech recognition tasks, namely the ISOLET database and TIDIGITS digit strings database. The SDP problem *Problem 5* in Section IV-E is solved by an open software, DSDP v5.6 [3] running under Matlab.

Algorithm 1: SDP Optimization

repeat

1) Perform Viterbi decoding for all training data based on models $\Lambda^{(n)}$
2) Identify the support set $\mathcal{S}$ according to **(2)**
3) An SDP algorithm is run to optimize the relaxed SDP *Problem 5*
4) Update models Λ based on the SDP solution Z^*: $\Lambda^{(n)} \Rightarrow \Lambda^{(n+1)}$
5) $n = n+1$

until some convergence conditions are met

In our experiments, we use the standard 39-dimension feature vectors, consisting of 12-D static MFCC, log-energy, delta, and acceleration coefficients. The LME training method is compared with the conventional maximum-likelihood estimation (MLE) and MCE. In the experiments, an MLE model is estimated based on the standard Baum–Welch algorithm. The best MLE model is used as the seed model to conduct the MCE training. Then, the best MCE model is used as the seed model for the LME training. All HMM model parameters are estimated during the MLE and MCE training. In the LME training, we only update Gaussian mean vectors while keeping other model parameters (including Gaussian covariance matrices and HMM transition probabilities) unchanged. During each epoch of LME training, the support token set is selected by setting γ to include exactly N (N ranging from 60 to 300) most competing support tokens. In this paper, we also compare two different optimization methods for LME, namely gradient descent method in [13] and the proposed SDP method.

A. Isolated Speech Recognition: ISOLET E-set Recognition

In our first set of experiments, the LME training based on the SDP method is evaluated on English E-set recognition with 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). An HMM recognizer with 16-state whole-

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

	1-mix	2-mix	4-mix
ML	85.56	90.56	91.48
MCE	91.48	94.07	93.89
LME-GD	**92.96**	95.00	94.44
LME-SDP	**92.96**	**95.19**	**95.00**

ML: Maximum-likelihood method. MCE: minimum classification error. LME-GD: LME method with gradient descent. LME-SDP: LME method with SDP.

word-based models is trained based on different training criteria. Here CDHMMs with 1-mix, 2-mix, and 4-mix per state are experimented. In this task, recognition error rate on the training data is quickly brought down to zero after a couple of iterations in MCE training, and it remains zero in the LME training on this task.

1) Performance of LME/SDP on E-set Recognition: 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, where LME-GD represents the LME with the constrained joint optimization method based on gradient descent search in [13] and [16], and LME-SDP represents the LME method with SDP proposed in this paper.

Experimental results in Table I clearly demonstrate that both LME methods work well on this task. For example, the models with 2-mix per state trained by the SDP method achieve the word accuracy of 95.19%, which indicates 18.89% errors reduction over the corresponding MCE-trained models, which get 94.07% in accuracy. And the models with 2-mix per state trained with gradient descent method achieves the word accuracy of 95.00%, which indicates 15.68% errors reduction over the corresponding MCE-trained models. 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 models are slightly over-trained in this small database. At last, if we compare the performance of two LME methods, we can see that SDP method performs slightly better than the gradient descent method on this task, especially in the case of 4-mix.

The range parameter r^2 in constraint (10) must be manually tuned to achieve the best performance. Fig. 1 plots the best word accuracy achieved by LME/SDP models under different normalized range parameter $r^2/\mathcal{K}$ (normalized by the total number of Gaussians in the HMMs) for 1-mix models on the E-set task. From the definition of the constraint, we know that the normalized range means how far each dimension of the mean vectors is allowed to move away from its original position in average. From the figure, we can see that with the increase of the normalized range, the word accuracy also increases until it reaches the highest value 92.96% at the position $r^2/\mathcal{K} = 0.1$; then, it begins to drop. It is easy to understand the process before the position $r^2/\mathcal{K} = 0.1$, since a larger constraint range should normally result in a better optimal SDP solution. The reason for the drop is due to the fact we adopt the Viterbi approximation in (6) to formulate the LME/SDP method. If the range is too large, HMM Gaussian means may be dramatically changed during the SDP optimization. It may invalidate the Viterbi approximation. In other words, the optimal Viterbi paths $\{\mathbf{s}^*, \mathbf{l}^*\}$, which we determined using the old HMMs prior to the SDP optimization, may not be dominant anymore if the HMMs significantly change during the SDP optimization. A different path may become dominant for the new updated model. This makes the SDP optimization irrelevant to the final speech recognition process since we have used the Viterbi approximation in (6) based on the original best path, $\{\mathbf{s}^*, \mathbf{l}^*\}$, to formulate the LME/SDP in the first place.

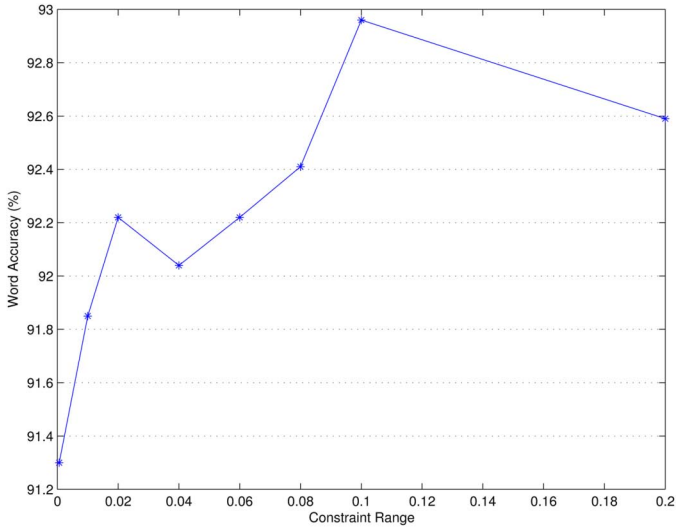


Fig. 1. Word accuracy of LME/SDP models on the test set is plotted as a function of the normalized range parameter $r^2/\mathcal{K}$ in the LME/SDP training of 1-mix models on the E-set recognition task.

TABLE II
BEST NORMALIZED RANGES FOR LME/SDP TRAINING

1-mix	2-mix	4-mix
0.1	0.04	0.02

The best normalized ranges for different models are shown in Table II. From the table, we can see that the optimal normalized range usually becomes small when mixture number is large. This is because all Gaussians become closer to each other as the mixture number increases. In this case, even a small change in the Gaussian means could make the initial Viterbi path not dominant anymore.

In our experiments, we have also investigated two different methods to update Gaussian means from the solution matrix Z^*, namely the projection method and the augmentation method. However, we have not observed any significant difference in recognition performance.

B. Continuous-Speech Recognition: TIDIGITS Digit Strings

In this section, the N-best based string-level LME/SDP algorithm has been evaluated in a connected digit string recognition task using the TIDIGITS corpus [14]. The corpus vocabulary is made of the digits "1" to "9," plus "oh" and "zero," for a total of 11 words. The length of the digit strings varies from 1 to 7 (except 6). Only the adult portion of the corpus is used in our experiments. The training set has 8623 digit strings,

TABLE III
TRAINING SET STRING ACCURACY (%) OF DIFFERENT MODELS

	ML	MCE	LME-GD	LME-SDP
1-mix	87.21	93.68	97.96	99.57
2-mix	95.08	96.27	99.35	99.90
4-mix	97.19	97.78	99.22	99.93
8-mix	98.61	98.76	99.51	99.95
16-mix	98.00	98.98	99.68	99.90
32-mix	99.26	99.41	99.74	99.90

TABLE IV
TEST SET STRING ERROR RATE (%) OF DIFFERENT MODELS

	ML	MCE	LME-GD	LME-SDP
1-mix	12.61	6.72	3.77 (44%)	2.75 (59%)
2-mix	5.26	3.94	1.70 (57%)	1.24 (69%)
4-mix	3.48	2.23	1.24 (44%)	0.89 (60%)
8-mix	1.94	1.41	0.87 (38%)	0.68 (52%)
16-mix	1.72	1.11	0.82 (26%)	0.63 (43%)
32-mix	1.34	0.90	0.66 (27%)	**0.53** (41%)

and the test set has 8700 strings. Our model set includes 11 whole-word CDHMMs representing all digits. Each HMM has 12 states and uses a simple left-to-right topology without state-skip. Different number of Gaussian mixture components (from 1 to 32 per HMM state) are experimented. The number of competing strings in the N-best list is experimentally set to five. On the TIDIGITS task, recognition accuracy on the training set is not 100% even after the MCE and LME training. The recognition result comparison on the training set between different training criteria are given in Table III for various model complexities, where LME-GD represents the LME with the constrained joint optimization method in [13] and [16], and LME-SDP represents the LME method with SDP proposed in this paper. 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 Tables IV and V, we give string error rates and word error rates in the test set for the best models obtained by different training criteria, respectively. The results are listed for various model sizes we have investigated. The results clearly show that the LME-SDP training method considerably reduces recognition error in terms of both string error rate and word error rate on top of the MCE training across all different model sizes. As the model size gets bigger, advantage of using LME decreases, but it still remains significant. For small model sizes (such as 1-mix, 2-mix, and 4-mix), the LME/SDP method yields over 50% relative string error reduction and 60% relative word error reduction on top of the MCE training. For large model sizes (such as 8-mix, 16-mix, and 32-mix), the SDP method gives around 30%–50% relative string and word error reduction. If we compare the performance with the LME-GD [13] methods, we can see that SDP method performs significantly better than the gradient descent-based method in terms of both string and word error rates. The SDP method shows its advantage to find the globally optimal solution (not just local optimum) in the augmented higher dimension space. Results also show that the approximation caused by relaxation and projection seems reasonably good in our experiments.

TABLE V
TEST SET WORD ERROR RATE (WER) (%) OF DIFFERENT MODELS. THE NUMBERS INSIDE PARENTHESES REPRESENT THE RELATIVE WORD ERROR REDUCTION OF LME OVER MCE

	MLE	MCE	LME-GD	LME-SDP
1-mix	4.55	2.40	1.28 (47%)	0.93 (61%)
2-mix	1.86	1.41	0.57 (60%)	0.42 (70%)
4-mix	1.18	0.76	0.43 (43%)	0.29 (62%)
8-mix	0.66	0.49	0.30 (39%)	0.23 (53%)
16-mix	0.57	0.38	0.29 (24%)	0.21 (45%)
32-mix	0.45	0.30	0.22 (27%)	**0.18 (40%)**

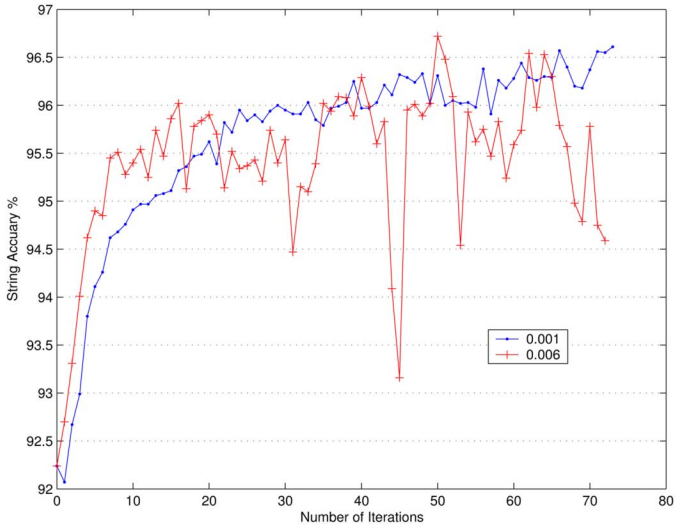


Fig. 2. Evolution of string accuracy for string-level LME/SDP training of 1-mix models under different normalized range parameters $r^2/\mathcal{K}$ on the TIDIGITS recognition task.

The range parameter r^2 in constraint (10) has been manually tuned for the best performance. Fig. 2 plots the evolution of the string accuracy on test data achieved by LME/SDP models under two different normalized range parameters $r^2/\mathcal{K}$ for 1-mixture models on the TIDIGITS task. From the figure, we can see that the range parameter plays an important role in the training processes. A small range value such as 0.001 in the figure will generate a relatively slow but stable training process, while a very large range value such as 0.006 in the figure will generate a relatively fast training process at the beginning, but it will cause the following steps unstable. Just as mentioned on the ISOLET E-set recognition task, a large range value may invalidate the Viterbi approximation, e.g., (8), which will makes the SDP optimization irrelevant to the speech recognition process. Thus, a suitable range value must be experimentally chosen to create a relatively stable process with acceptable converging speed. The normalized range is set between 0.001 and 0.004 for

the LME/SDP training of the models with various sizes in our experiments.

As far as computational complexity is concerned, it is still quite expensive to solve a large-scale SDP problem. As a result, the LME/SDP training takes much longer time than the gradient descent method, especially for large model sets. However, the LME/SDP tends to converge to a better solution due to its convex optimization nature.

VI. Conclusion

In this paper, we have proposed an SDP method for LME of CDHMMs in speech recognition. At first, we have proposed a new locality-based constraint for LME of CDHMMs in speech recognition. The new LME problem subject to this constraint can be transformed to an SDP problem under some relaxation conditions. Then, the LME training of CDHMMs can be sloved with efficient SDP optimization algorithms. We have investigated its performance on a speaker-independent English E-set isolated-word recognition task using the ISOLET database and a speaker-independent continuous digit string recognition task using the TIDIGITS database. The newly proposed LME/SDP method has been demonstrated to be quite effective on both tasks. Currently, the LME/SDP method is being extended to other large-vocabulary continuous-speech recognition speech recognition tasks. In our future work, we will analyze the theoretical convergence property of the proposed LME/SDP algorithm as in [25].

References

[1] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *Soc. Industrial Appl. Math. (SIAM) J. Optim.*, vol. 5, pp. 13–51, 1995.

[2] Y. Altun, I. Tsochantaridis, and T. Hofmann, "Hidden Markov support vector machines," in *Proc. 20th Int. Conf. Mach. Learn. (ICML-2003)*, Washington, D.C, 2003, pp. 1–8.

[3] S. J. Benson, Y. Ye, and X. Zhang, "Solving large-scale sparse semidefinite programs for combinatorial optimization," *Soc. Industrial Appl. Math. (SIAM) J. Optim.*, vol. 10, no. 2, pp. 443–461, 2000.

[4] P. Biswas and Y. Ye, "Semidefinite programming for ad hoc wireless sensor network localization," in *Proc. Inf. Process. Sensor Netw. (IPSN)*, Berkeley, CA, Apr. 2004, pp. 46–54.

[5] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, "Linear matrix inequalities in system and control theory," *Stud. Appl. Math. (SIAM)*, vol. 15, Jun. 1994.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[7] M. Fukuda and M. Kojima, "Interior-Point Methods for Lagrangian Duals of Semidefinite Programs," Dept. Math. Comput. Sci., Tokyo Inst. Technol., Tokyo, Japan, 2000, Tech. Rep. B-365.

[8] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *J. ACM*, vol. 42, pp. 1115–1145, 1995.

[9] D. Goldfarb and K. Scheinberg, "Interior point trajectories in semidefinite programming," *Soc. Industrial Appl. Math. (SIAM) J. Optim.*, vol. 8, pp. 871–886, 1998.

[10] 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.

[11] H. Jiang, Discriminative Training for Large Margin HMMs Dept. Comput. Sci. Eng., York Univ., Toronto, ON, Canada, 2004, Tech. Rep. CS-2004–01.

[12] 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.

[13] H. Jiang, X. Li, and C. Liu, "Large margin hidden Markov models for speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1584–1595, Sep. 2006.

[14] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proc. ICASSP'84*, 1984, pp. 328–331.

[15] X. Li, H. Jiang, and C. Liu, "Large margin HMMs for speech recognition," in *Proc. 2005 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'05)*, Philadelphia, PA, Mar. 2005, pp. V-513–V-516.

[16] X. Li and H. Jiang, "A constrained joint optimization method for large margin HMM estimation," in *Proc. IEEE Workshop Autom. Speech Recognition Understanding*, 2005, pp. 151–156.

[17] Y. Nesterov and A. Nemirovskii, "Interior-point polynomial algorithms in convex programming," in *SIAM Studies in Applied Mathematics*. Philadelphia, PA: SIAM, 1994, vol. 13.

[18] X. Li, "Large margin hidden markov models for speech recognition," M.S thesis, Dept. Comput. Sci. Eng., York Univ., Toronto, ON, Canada, 2005.

[19] X. Li and H. Jiang, "Solving large margin HMM estimation via semi-definite programming," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP'06)*, Pittsburgh, PA, Apr. 2006, pp. 2414–2417.

[20] 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'05)*, Philadelphia, PA, Mar. 2005, pp. V-101–V-104.

[21] 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. 2005 IEEE Workshop Autom. Speech Recognition Understanding*, 2005, pp. 420–425.

[22] F. Sha and L. Saul, "Large margin hidden Markov models for automatic speech recognition," *Adv. Neural Inf. Process. Syst. (NIPS) 19*, pp. 1249–1256, 2006.

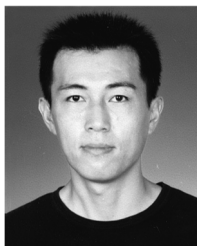
[23] A. M. So and Y. Ye, "Theory of semidefinite programming for sensor network localization," Dept. Manag. Sci. Eng., Stanford Univ., Stanford, CA, 2004, Tech. Rep..

[24] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," in *Proc. Neural Inf. Process. Syst. Conf. (NIPS'03)*, Vancouver, BC, Canada, Dec. 2003.

[25] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proc. 21th Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 104–112.

[26] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev. 38*, pp. 49–95, Mar. 1996.

[27] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.



Xinwei Li recieved 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, Inc. His major research interest focuses on automatic speech recognition especially discriminative training.

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

From October 1998 to April 1999, he worked as a Researcher in the University of Tokyo. From April 1999 to June 2000, he was with Department of Electrical and Computer Engineering, University of Waterloo, Waterloom ON, Canada, as a Postdoctoral Fellow. From 2000 to 2002, he worked in Dialogue Systems Research, Multimedia Communication Research Lab, Bell Labs, Lucent Technologies, Inc., Murray Hill, NJ. He joined Department of Computer Science and Engineering, York University, Toronto, ON, Canada, as an Assistant Professor in fall 2002 and was promoted to Associate Professor in 2007. 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.