Large-Margin Estimation of Hidden Markov Models With Second-Order Cone Programming for Speech Recognition

Dalei Wu, Yan Yin, and Hui Jiang, *Member, IEEE*

***Abstract*—Large-margin estimation (LME) holds a property of good generalization on unseen test data. In our previous work, LME of HMMs has been successfully applied to some small-scale speech recognition tasks, using the SDP (semi-definite programming) technique. In this paper, we further extend the previous work by exploring a more efficient convex optimization method with the technique of second-order cone programming (SOCP). More specifically, we have studied and proposed several SOCP relaxation techniques to convert LME of HMMs in speech recognition into a standard SOCP problem so that LME can be solved with more efficient SOCP methods. The formulation is general enough to deal with various types of competing hypothesis space, such as N-best lists and word graphs. The proposed LME/SOCP approaches have been evaluated on two standard speech recognition tasks. The experimental results on the TIDIGITS task show that the SOCP method significantly outperforms the gradient descent method, and achieve comparable performance with SDP, but with 20–200 times faster speed, requiring less memory and computing resources. Furthermore, the proposed LME/SOCP method has also been successfully applied to a large vocabulary task using the Wall Street Journals (WSJ0) database. The WSJ-5k recognition results show that the proposed method yields better performance than the conventional approaches including maximum-likelihood estimation (MLE), maximum mutual information estimation (MMIE), and more recent boosted MMIE methods.**

***Index Terms*—Convex optimization, convex relaxation, discriminative training (DT), large-margin estimation (LME), second-order cone programming (SOCP).**

I. Introduction

AUTOMATIC speech recognition (ASR) has been one of the most challenging tasks in the field of pattern recognition. Many researchers have been contributing their efforts to the research of ASR for several decades. Summarizing these efforts, the most successful modeling methods that are widely accepted by the ASR research community can be organized into two categories: generative training (GT) and discriminative training (DT). In generative training methods, ASR model parameters are estimated using a standard training algorithm that is referred to as maximum-likelihood estimation (MLE) under an assumption that all the training data can be "generated" from the presumed distributions of speech models, whereas in DT methods, ASR model parameters are optimized based on a training data set to maximize or minimize a certain discriminative criterion. The most popular acoustic model for ASR is continuous density Gaussian mixture hidden Markov models (CDHMMs) and a variety of DT methods have been investigated for CDHMMs, such as maximum mutual information estimation (MMIE) [2], [23], minimum classification error (MCE) [4], [8], [9], [19], [20], minimum word/phone error (MWE/MPE) [23].

Although most of these methods are quite effective, one of the notable drawbacks is that they do not possess an attribute to avoid the well-known over-fitting problem, i.e., the models trained on a given training set may not be well generalized to unseen test data. From a theoretical point of view in machine learning, a large-margin classifier implies good generalization power and generally yields much lower generalization errors on unseen data. More recently, large-margin based discriminative training methods have been successfully applied to speech recognition, where Gaussian mixture continuous density hidden Markov models (CDHMMs) are estimated based on the principle of maximizing the minimum margin, such as [11] and [12], where the large-margin estimation (LME) of HMMs turns out to be a constrained minimax optimization problem. However, the major difficulty of applying the LME method, as well as other DT criteria, to large-scale ASR tasks lies in the fact that these DT criteria normally result in large-scale optimization problems and how to solve these problems is very challenging [13]. For instance, discriminative training of a state-of-the-art system in large-scale ASR tasks normally needs to solve an optimization problem involving over millions or even tens of millions of free parameters. Optimizing a complicated objective function in such a high-dimension space is an extremely difficult task since optimization can be easily trapped to any shallow local optimal point of the function surface. Most of the optimization methods used by DT in ASR suffer from this problem, including so far the most popular optimization method derived from a set of growth functions, i.e., extended Baum–Welch (EBW) algorithm. One

Manuscript received November 17, 2009; revised March 06, 2010; accepted November 07, 2010. Date of publication December 03, 2010; date of current version June 01, 2011. This work was supported in part by a Natural Sciences and Engineering Research Council of Canada (NSERC) discovery grant. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark Gales.

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

Y. Yin was with the Department of Computer Science and Engineering, York University, Toronto, ON M3J 1P3, Canada. He is now with Li Creative Technologies, Inc., NJ 07932 USA (e-mail: yyin@licreativetech.com).

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.2010.2096213

possible way to solve this problem is to use convex optimization methods since any locally optimal point in a convex optimization problem is guaranteed to be globally optimal. In our previous work, we have successfully applied one of the common convex optimization methods, namely semi-definite programming (SDP), to solve LME optimization problems based on N-best lists for some small-vocabulary digit string recognition tasks [16], [29]. More specifically, large-margin estimation (LME) of Gaussian mixture CDHMMs has been formulated as a semi-definite programming (SDP) problem under various SDP relaxation conditions as in [16] and [29]. As a result, the LME problem can be solved by many popular convex optimization algorithms, such as interior-point methods, which lead to the globally optimal solution, because SDP is a well-defined convex optimization problem. Moreover, it has been experimentally shown that the SDP relaxation is extremely tight to maintain high accuracy for LME. Therefore, it has been reported in [16] that the SDP-based large-margin estimation method (denoted as LME/SDP for short) outperforms all other discriminative training methods used for speech recognition and it has achieved one of the best performances in the standard TIDIGITS connected digit string speech recognition task. However, optimization time of the LME/SDP method increases dramatically as the size of HMM models grows because the size of the SDP variable matrix in [16] is roughly equal to the square of the total number of Gaussians in the model set. In [16], the LME/SDP method has been successfully managed to handle a CDHMM set consisting of about 4 k Gaussians but it is unlikely to directly extend the LME/SDP method in [16] to any large-vocabulary speech recognition task which typically involves tens or even hundreds of thousands of Gaussians due to extensive CPU time and a large amount of memory required for solving such a large-scale SDP problem.

In this paper, we propose to use a different convex optimization method, namely second-order cone programming (SOCP), to solve the large-margin estimation of CDHMMs for speech recognition. Comparing with SDP, SOCP is a simpler convex optimization problem and SOCP can be solved much faster than SDP for the same problem size and structure; see [15] and [18], but just like SDP, an SOCP algorithm can guarantee to find the globally optimal solution since SOCP is also a well-defined convex optimization problem. In order to apply LME to a large-scale ASR tasks, in this paper, we extend the previous work by formulating LME of CDHMMs into a convex quadratic optimization problem based on a general representation of competing hypothesis space, which includes both N-best lists for small-scale ASR tasks and word graphs for large-vocabulary ASR tasks. Based on the SOCP relaxation method originally proposed in [15], we have formulated large-margin estimation (LME) of CDHMMs as an SOCP problem, where the size of SOCP variable vector is only proportional to the total number of Gaussians, as opposed to the square of the number of Gaussians in the LME/SDP method. In this way, the proposed LME method (denoted as LME/SOCP) can deal with much larger HMM model sets widely used in large-scale ASR tasks. However, it has been experimentally found that the original SOCP relaxation in [15] is too loose to yield as good performance as a gradient descent or SDP method for LME in speech recognition (cf. Table I). In this paper, we have proposed two novel tighter SOCP relaxation methods for LME of CDHMMs. Experimental results on the standard TIDIGITS task show that the LME/SOCP methods based on the newly proposed SOCP relaxations significantly outperform the previous gradient descent method and they can achieve almost comparable performance as the previous LME/SDP approach, but the LME/SOCP method shows much better efficiency in terms of optimization time (about 20–200 times faster) and memory usage when compared with the LME/SDP method in [16]. Furthermore, the LME/SOCP method based on the new SOCP relaxation technique has been successfully applied to a large vocabulary continuous speech recognition (LVCSR) task using the standard WSJ-5k corpus. Experimental results have shown the LME/SOCP method significantly outperforms the conventional maximum-likelihood estimation (MLE) method as well as two popular discriminative training methods, i.e., MMIE and boosted MMIE methods using the standard EBW based optimization algorithm.

It is also worth to note that there have been some recent research efforts which attempt to include a margin term into the conventional DT training framework of ASR, e.g., boosted MMI/Modified MMI, boosted MPE/MWE [7], [25], margin-based MMI/MPE [21], etc. The idea of these methods is to introduce some extra parameters to the numerator and/or denominator of the conventional MMI/MPE criteria to approximate the hinge loss function of the hidden Markov support vector machine (HM-SVM) [1]. Roughly speaking, these margin-based MMI criteria can also be regarded as a special case of LME with the hinge loss function as used in HM-SVM. These methods normally reply on the conventional EBW-based method to optimize their objective functions.

The remainder of this paper is organized as follows: In Section II, we first describe the framework of large-margin estimation for ASR. In Section III, we briefly introduce second-order cone programming (SOCP) as a special case of convex optimization. In Section IV, we show how to convert the LME problem into an SOCP problem and present three different SOCP relaxation methods. Experiments on both TIDIGITS and WSJ-5k are reported and discussed in Section V. Finally, the paper is concluded with our findings and discussions in Section VI.

II. Large-Margin Estimation of HMMS

Building a standard ASR system normally consists of two separate stages: training and test. At the training stage, a set of acoustic models, normally CDHMMs, denoted as Λ, is estimated to represent basic speech units, such as phonemes, bi-phones, tri-phones, etc., under a certain training criterion. At the test stage, the trained CDHMM set Λ is then used to search the best-matched word sequence for each acoustic observation X by maximizing the posterior probability of a word sequence given the observation X and the trained model set Λ

$$\begin{aligned} W^* &= \arg\max_w \; p(w|X,\Lambda) \\ &= \arg\max_w \; p(w)\cdot p(X|w,\Lambda) \\ &= \arg\max_w \; \mathcal{F}(X|w,\Lambda) \end{aligned} \tag{1}$$

where $p(w)$ is language model score, normally calculated from n-grams language models, $p(X|w,\Lambda)$ is acoustic score obtained based on CDHMMs Λ, and $\mathcal{F}(X|w,\Lambda) = \log[p(w)\cdot p(X|w,\Lambda)]$ is referred to as discriminant function in the logarithm domain.

The formulation of the large-margin criterion depends on an essential concept of *multiclass separation margin*. Given a sentence X_r, $r\in[1,R]$, in a training set $\mathcal{D}$ along with its correct transcription W_r, we define a competing hypothesis space for the sentence X_r, which is normally generated from a decoding process, as $\mathcal{H}_r$. In the LME formulation, competing hypothesis space $\mathcal{H}_r$ always *excludes* the correct transcription W_r. Following [1], [5], and [26], the *mutliclass separation margin* for sentence X_r is defined as the minimum distance between correct transcription W_r and competing hypotheses w_r, i.e.,

$$d_r(\Lambda) = \mathcal{F}(X_r|W_r,\Lambda) - \max_{w_r\in\mathcal{H}_r}\mathcal{F}(X_r|w_r,\Lambda) \tag{2}$$

where $\mathcal{F}(X_r|W_r,\Lambda)$ and $\mathcal{F}(X_r|w_r,\Lambda)$ are discriminant functions for the correct path W_r and a competing hypothesis w_r. They are normally calculated as logarithm of the product of acoustic and LM scores as follows:

$$\mathcal{F}(X_r|W_r,\Lambda) = \log[p(W_r)\cdot p(X_r|W_r,\Lambda)] \tag{3}$$
$$\mathcal{F}(X_r|w_r,\Lambda) = \log[p(w_r)\cdot p(X_r|w_r,\Lambda)] \tag{4}$$

where $p(W_r)$, $p(w_r)$ are language model scores and $p(X_r|W_r,\Lambda)$, $p(X_r|w_r,\Lambda)$ are acoustic scores for the correct transcript W_r and the competing hypotheses w_r, respectively.

By substituting (3) and (4) into (2), we can rewrite the margin distance of sentence X_r as follows:

$$\begin{aligned} d(X_r|\Lambda) &= \mathcal{F}(X_r|W_r,\Lambda) - \max_{w_r\in\mathcal{H}_r}\mathcal{F}(X_r|w_r,\Lambda) \\ &= \log[p(W_r)\cdot p(X_r|W_r,\Lambda)] \\ &\quad - \max_{w_r\in\mathcal{H}_r}\log[p(w_r)\cdot p(X_r|w_r,\Lambda)]. \end{aligned} \tag{5}$$

Since *max* function is not smooth, in practice, it is often substituted for by an alternative, *soft-max*, using the operator of *log-sum*. Therefore, the final definition of margin distance $d(X_r|\Lambda)$ for the sentence X_r is derived as follows:

$$\begin{aligned} d(X_r|\Lambda) = \log[p(W_r)\cdot p(X_r|W_r,\Lambda)] \\ - \log\sum_{w_r\in\mathcal{H}_r}[p(w_r)\cdot p(X_r|w_r,\Lambda)]. \end{aligned} \tag{6}$$

In practice, a scaling factor κ $(0<\kappa<1)$ is often introduced into calculating discriminant functions to prevent some hypotheses from dominating the sum, i.e.,

$$\mathcal{F}(X_r|W_r,\Lambda) = \log[p(W_r)^\kappa\cdot p(X_r|W_r,\Lambda)^\kappa] \tag{7}$$
$$\mathcal{F}(X_r|\mathcal{H}_r,\Lambda) = \log\sum_{w_r\in\mathcal{H}_r}[p(w_r)^\kappa\cdot p(X_r|w_r,\Lambda)^\kappa] \tag{8}$$

where $\mathcal{F}(X_r|\mathcal{H}_r,\Lambda)$ denote the discriminant function calculated from the whole hypothesis space $\mathcal{H}_r$.

With all these necessary notations, the criterion of LME can be then defined as an optimization procedure to seek an optimal model set Λ^* by maximizing the minimum margin distance over all sentences in the training set $\mathcal{D}$, i.e.,

$$\Lambda^* = \arg\max_{\Lambda}\min_{X_r\in\mathcal{D}} d(X_r|\Lambda) \tag{9}$$

where $d(X_r|\Lambda)$ denotes the margin of a sentence X_r calculated based on the model set Λ.

The above LME formulation is applicable to the case where there is no training error and all training sentences have positive margins, i.e., $d(X_r|\Lambda)>0$. It can be easily extended to a more general case to consider training errors, as in [12]. In this case, we first select two subsets from the overall training set: the support token set $\mathcal{S}$ and the error set $\mathcal{E}$. The support token set $\mathcal{S}$ contains only positive tokens with relatively small margins while the error set $\mathcal{E}$ includes all negative tokens with negative margins. In other words, the support token set $\mathcal{S}$ is denoted as

$$\mathcal{S} = \{X_r : 0\le d_r(\Lambda)\le d, r\in[1,R]\} \tag{10}$$

which contains all sentences in the training set with relatively small positive margin less than a threshold d. The error set $\mathcal{E}$ is denoted as

$$\mathcal{E} = \{X_r : d_r(\Lambda)<0, r\in[1,R]\} \tag{11}$$

which then includes all sentences with negative margins. Conceptually speaking, the support token set $\mathcal{S}$ includes all training sentences locating in the correct decision region but staying quite close to the current decision boundary while the error token set $\mathcal{E}$ includes all training sentences locating in the wrong decision region.

With the concepts of support and error tokens, training errors can be taken into consideration to facilitate the LME training and thus, a variant criterion of LME, referred to as soft large-margin estimation (soft-LME) [12], can be defined as follows:

$$\Lambda^* = \arg\max_{\Lambda}\left[\min_{X_r\in\mathcal{S}} d(X_r|\Lambda) + \frac{\epsilon}{|\mathcal{E}|}\sum_{X_r\in\mathcal{E}} d(X_r|\Lambda)\right] \tag{12}$$

where ϵ is a constant to balance minimum margin with training errors, and $|\mathcal{E}|$ is the number of error tokens contained in a training set.

In addition to the objective function of LME as established in (12), an additional locality constraint is often imposed during the optimization process to control the updating range of model parameters Λ, as shown in [17], in order to avoid changing the new model parameters too much from the current location. The locality constraint is simply defined as Kullback–Leibler (KL) divergence of the model parameters Λ and its current values, i.e.,

$$\mathcal{D}(\Lambda|\Lambda^{(n)}) = \sum_{\lambda\in\Lambda}\mathcal{D}(\lambda\|\lambda^{(n)})\le h^2 \tag{13}$$

where h is a constant to control the range for updating model parameters.

By considering both (12) and the locality constraint of (13), if we further introduce two variables ρ and ρ' to represent the lower bound of the margins of all support tokens and the upper bound of the sum of the margins of all error tokens, respectively,

then as in [16], [28], we can equivalently convert the maximin optimization problem in (12) into an constrained minimization problem as follows.

1) Problem 1:

$$\min_{\Lambda,\rho,\rho'}\left[-\rho+\epsilon\cdot\frac{1}{|\mathcal{E}|}\cdot\rho'\right] \tag{14}$$

$$\text{subject to:}\quad \mathcal{F}(X_r|\mathcal{H}_r,\Lambda)-\mathcal{F}(X_r|W_r,\Lambda)\le-\rho \text{ for all } X_r\in\mathcal{S} \tag{15}$$

$$\sum_{X_r\in\mathcal{E}}[\mathcal{F}(X_r|\mathcal{H}_r,\Lambda)-\mathcal{F}(X_r|W_r,\Lambda)]\le\rho' \tag{16}$$

$$\mathcal{D}(\Lambda|\Lambda^{(n)})=\sum_{\lambda\in\Lambda}\mathcal{D}(\lambda\|\lambda^{(n)})\le h^2 \tag{17}$$

$$\rho\ge 0 \text{ and } \rho'\ge 0. \tag{18}$$

Until now, we have formulated LME of HMMs into a general constrained optimization problem. Hence, the next question would be how to efficiently solve the formulated optimization problem. In Sections III–VI, we will present our ideas to use SOCP to efficiently solve the above general optimization problem for ASR.

III. Second-Order Cone Programming

In this section, let us first briefly introduce second-order cone programming (SOCP) as a special case of convex optimization. An SOCP problem is a nonlinear convex optimization problem in which a linear function is minimized over intersection of an affine set and various second-order cone constraints. A standard SOCP has the following form:

$$\text{minimize}\quad f^Tx \tag{19}$$

$$\text{subject to}\quad \|A_ix+b_i\|\le c_i^Tx+d_i\quad(i=1,\dots,N) \tag{20}$$

where $x\in\mathbf{R}^n$ is the optimization variable, and the problem parameters include $f\in\mathbf{R}^n$, $A_i\in\mathbf{R}^{(n_i-1)\times n}$, $b_i\in\mathbf{R}^{n_i-1}$, $c_i\in\mathbf{R}^n$, and $d_i\in\mathbf{R}$. The norm appearing in the constraints is the standard Euclidean norm, i.e., $\|u\|=(u^Tu)^{1/2}$. We call the constraint $\|A_ix+b_i\|\le c_i^Tx+d_i$ in (19) as second-order convex cone constraint of dimension n_i.

SOCP includes linear programming (LP) and convex quadratic programs (QP) as special cases, but is less general than SDP. Many efficient primal-dual interior-point methods have been developed for SOCP. Like LP and QP and SDP, an SOCP problem can be solved in polynomial time by interior point methods. The computational effort per iteration required by these methods to solve SOCP problems, although greater than LP and QP problems, is much less than that required to solve an SDP problem with similar size and structure. Since SOCP is a well-defined convex optimization problem, the efficient algorithm for SOCP can always lead to the globally optimal solution.

IV. General Solution to LME via SOCP

In this section, we propose a general framework to use the SOCP algorithm to efficiently solve the optimization problem in LME of HMMs, i.e., *Problem* 1, using several SOCP relaxation techniques. The method is derived based on a general representation of competing hypothesis space, which can be easily adapted to either N-best list for small scale ASR tasks or word graph for LVCSR.

In our proposed method, we only consider to optimize mean vectors of Gaussian mixture CDHMMs and leave other parameters unchanged. We assume totally $\mathcal{K}$ Gaussian mixtures in the CDHMM set, each of which is denoted by $\mathcal{N}(\mu_k,\Sigma_k)$ with $k\in[1,\dots,\mathcal{K}]$. For simplicity, all covariance matrices are assumed to be diagonal as $\Sigma_k=\text{diag}\left(\sigma_{k1}^2,\dots,\sigma_{kD}^2\right)$.

From *Problem* 1, we can see that equations for both support and error tokens are mostly determined by two terms, $\mathcal{F}(X_r|W_r,\Lambda)$ and $\mathcal{F}(X_r|\mathcal{H}_r,\Lambda)$. The former term is related to correct transcription W_r, which is normally referred to as the *numerator* term and the latter one is related to all competing hypotheses in the space $\mathcal{H}_r$, which is called the *denominator* term. Next, we will show how to convert these two terms into quadratic forms, which are then used to convert *Problem* 1 into an SOCP problem.

Let us first visit the numerator term $\mathcal{F}(X_r|W_r,\Lambda)$. This term can be approximated by an auxiliary function as in the well-known expectation–maximization (EM) algorithm [6], which is referred to as expectation-based approximation (E-approx) methods in [14]. By using E-approx, we can approximate the numerator term $\mathcal{F}(X_r|W_r,\Lambda)$ in (15) with an auxiliary function, denoted as $Q_r^+(\Lambda|\Lambda^{(n)})$, with the following form:

$$\begin{aligned}\mathcal{F}(X_r|W_r,\Lambda)&\approx Q_r^+(\Lambda|\Lambda^{(n)})\\&=\sum_{\mathbf{s}_r,\mathbf{l}_r}[\log p(X_r,\mathbf{s}_r,\mathbf{l}_r\mid\Lambda_{W_r})+\log p(W_r)]\\&\quad\cdot\Pr\left(\mathbf{s}_r,\mathbf{l}_r\mid X_r,\Lambda_{W_r}^{(n)}\right)\\&=-\frac{1}{2}\sum_{t=1}^{T_r}\sum_{k\in\mathcal{K}}\sum_{d=1}^{D}\frac{(x_{rtd}-\mu_{kd})^2}{\sigma_{kd}^2}\\&\quad\cdot\gamma_r(k,t)+b_r^*\end{aligned} \tag{21}$$

where $\mathbf{s}_r$ is a hypothesized state sequence and $\mathbf{l}_r$ is a hypothesized Gaussian sequence obtained in decoding sentence X_r, $Pr\left(\mathbf{s}_r,\mathbf{l}_r\mid X_r,\Lambda_{W_r}^{(n)}\right)$ is posterior probability for state and Gaussian sequence $(\mathbf{s}_r,\mathbf{l}_r)$, $\gamma_r(k,t)$ denotes the posterior probability calculated for the kth Gaussian component in the model set $(k\in\mathcal{K})$ at time t using the Baum–Welch algorithm from the correct path of utterance X_r conditional on the initial model $\Lambda^{(n)}$, and b_r^* is a constant independent from all Gaussian mean vectors calculated as follows:

$$\begin{aligned}b_r^*=&\sum_{k\in\mathcal{K}}\sum_{t=1}^{T_r}\log\pi_k\cdot\gamma_r(k,t)+\sum_{k\in\mathcal{K}}\sum_{t=1}^{T_r}\log w_k\cdot\gamma_r(k,t)\\&-\frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{t=1}^{T_r}\sum_{d=1}^{D}\log\sigma_{kd}^2\cdot\gamma_r(k,t).\end{aligned} \tag{22}$$

where π_k and w_k are initial probability and weight for the kth Gaussian and σ_{kd} is the d-dimensional variance in the kth Gaussian of HMM model set Λ.

After rearranging all terms in (21), we can organize $\mathcal{Q}_r^+(\Lambda|\Lambda^{(n)})$ as a quadratic function of all Gaussian mean vectors as

$$\mathcal{Q}_r^+(\Lambda|\Lambda^{(n)}) = -\frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{d=1}^{D} \xi_{rk}\cdot\left(\frac{\mu_{kd}-\bar{x}_{rkd}}{\sigma_{kd}}\right)^2 + b'_r \quad (23)$$

where b'_r is another constant independent of Gaussian mean vectors, and ξ_{rk} and $\bar{x}_{rkd}$ are statistics collected from the correct path of utterance X_r for the kth Gaussian component as follows:

$$b'_r = b^*_r - \frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{d=1}^{D}\xi_{rk}\left(\frac{\overline{x^2_{rkd}}-\bar{x}^2_{rkd}}{\sigma^2_{kd}}\right) \quad (24)$$

$$\xi_{rk} = \sum_{t=1}^{T_r}\gamma_r(k,t) \quad (25)$$

$$\bar{x}_{rkd} = \frac{\sum_{t=1}^{T_r}\gamma_r(k,t)\cdot x_{rtd}}{\sum_{t=1}^{T_r}\gamma_r(k,t)} \quad (26)$$

$$\overline{x^2_{rkd}} = \frac{\sum_{t=1}^{T_r}\gamma_r(k,t)\cdot x^2_{rtd}}{\sum_{t=1}^{T_r}\gamma_r(k,t)}. \quad (27)$$

Furthermore, we can represent (23) using standard quadratic functions as in the following matrix form:

$$Q_r^+(\Lambda|\Lambda^{(n)}) = -\frac{1}{2}\left[\mathbf{u}^T\Phi_r^+\mathbf{u} - 2\tilde{\mathbf{x}}_r^T\Phi_r^+\mathbf{u} + \tilde{\mathbf{x}}_r^T\Phi_r^+\tilde{\mathbf{x}}_r\right] + b'_r \quad (28)$$

where $\mathbf{u}$ is a large column vector created by concatenating all normalized Gaussian mean vectors as

$$\mathbf{u} = (\tilde{\mu}_1,\ldots,\tilde{\mu}_{\mathcal{K}}) \quad (29)$$

with $\tilde{\mu}_k = (\mu_{k1}/\sigma_{k1},\ldots,\mu_{kD}/\sigma_{kD})$, and Φ_r^+ is $\mathcal{K}D\times\mathcal{K}D$ dimensional diagonal matrix with all ξ_{rk} calculated in (25) as its diagonal elements, and $\tilde{\mathbf{x}}_r$ is a concatenated vector as $\tilde{\mathbf{x}}_r = (\tilde{\mathbf{x}}_{r1},\ldots,\tilde{\mathbf{x}}_{r\mathcal{K}})$ with $\tilde{\mathbf{x}}_{rk} = (\bar{x}_{rk1}/\sigma_{rk1},\ldots,\bar{x}_{rkD}/\sigma_{kD})$, where $\bar{x}_{rkd}$ is the d-dimensional statistics vector collected for kth Gaussian from all feature vectors in the sentence X_r, as in (26).

Next, let us consider the denominator term $\mathcal{F}(X_r|\mathcal{H}_r,\Lambda)$ for competing hypotheses. As we have mentioned before, all competing hypotheses w_r are encoded in a competing space $\mathcal{H}_r$. While N-best list is an appropriate representation for $\mathcal{H}_r$ in small scale ASR tasks, word graph is a more efficient format for $\mathcal{H}_r$ to represent competing hypotheses in LVCSR, due to extremely large number of different competing hypotheses in LVCSR.

Using the similar approximation strategy of E-approx, the denominator term $\mathcal{F}(X_r|\mathcal{H}_r,\Lambda)$ in (15) and (16) can also be formulated as a quadratic form of Gaussian mean vectors, as follows:

$$\begin{aligned}
&\mathcal{F}(X_r|\mathcal{H}_r,\Lambda)\\
&\approx \mathcal{Q}_r^-(\Lambda|\Lambda^{(n)})\\
&= \sum_{w\in\mathcal{H}_r}\sum_{\mathbf{s}_r,\mathbf{l}_r}[\log p(X_r,\mathbf{s}_r,\mathbf{l}_r|\Lambda_w)\\
&\qquad + \log p(w)]\cdot\Pr\left(\mathbf{s}_r,\mathbf{l}_r|X_r,\Lambda_w^{(n)}\right)\\
&= -\frac{1}{2}\sum_{w\in\mathcal{H}_r}\sum_{t=1}^{T_r}\sum_{k\in\mathcal{K}}\sum_{d=1}^{D}\left[\frac{(x_{rtd}-\mu_{kd})^2}{\sigma^2_{kd}}\cdot\gamma'_r(k,t|w)\right]\\
&\quad + \hat{b}_r\\
&= -\frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{d=1}^{D}\left[\xi'_{rk}\cdot\left(\frac{\bar{x}'_{rkd}-\mu_{kd}}{\sigma_{kd}}\right)^2\right] + b''_r
\end{aligned} \quad (30)$$

where $\mathbf{s}_r$ is a hypothesized state sequence and $\mathbf{l}_r$ is a hypothesized Gaussian sequence obtained in decoding sentence X_r, $\Pr\left(\mathbf{s}_r,\mathbf{l}_r \mid X_r,\Lambda_w^{(n)}\right)$ is posterior probability for the state and Gaussian sequence $(\mathbf{s}_r,\mathbf{l}_r)$, (μ_{kd},σ_{kd}) are the d-dimensional components of mean and variance for the kth Gaussian, b''_r, and $\hat{b}_r$ are constants

$$\begin{aligned}
\hat{b}_r = &\sum_{w\in\mathcal{H}_r}\sum_{k\in\mathcal{K}}\sum_{t=1}^{T_r}\log\pi_k\cdot\gamma'_r(k,t|w)\\
&+\sum_{w\in\mathcal{H}_r}\sum_{k\in\mathcal{K}}\sum_{t=1}^{T_r}\log w_k\\
&\cdot\gamma'_r(k,t|w) - \sum_{w\in\mathcal{H}_r}\frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{t=1}^{T_r}\sum_{d=1}^{D}\log\sigma^2_{kd}\\
&\cdot\gamma'_r(k,t|w)
\end{aligned} \quad (31)$$

$$b''_r = \hat{b}_r - \frac{1}{2}\sum_{k\in\mathcal{K}}\sum_{d=1}^{D}\xi_{rk}\left(\frac{\overline{x^2_{rkd}}'-\bar{x}^2_{rkd}}{\sigma^2_{kd}}\right) \quad (32)$$

and $\gamma'_r(k,t|w)$ is posterior probability for the kth Gaussian at time t calculated for any single competing hypothesis w in the hypothesis space $\mathcal{H}_r$ given the sentence X_r, and ξ'_{rk} and $\bar{x}'_{rkd}$ are statistics collected from $\mathcal{H}_r$ for the sentence X_r as follows:

$$\xi'_{rk} = \sum_{w\in\mathcal{H}_r}\sum_{t=1}^{T_r}\gamma'_r(k,t|w) \quad (33)$$

$$\bar{x}'_{rkd} = \frac{\sum_{w\in\mathcal{H}_r}\sum_{t=1}^{T_r}\gamma'_r(k,t|w)\cdot x_{rtd}}{\sum_{r=1}^{R}\sum_{w\in\mathcal{H}_r}\sum_{t=1}^{T_r}\gamma'_r(k,t|w)} \quad (34)$$

$$\overline{x^2_{rkd}}' = \frac{\sum_{w\in\mathcal{H}_r}\sum_{t=1}^{T_r}\gamma'_r(k,t|w)\cdot x^2_{rtd}}{\sum_{w\in\mathcal{H}_r}\sum_{t=1}^{T_r}\gamma'_r(k,t|w)}. \quad (35)$$

In the same way, we can represent $Q_r^-(\Lambda|\Lambda^{(n)})$ as a standard quadratic form in matrix form

$$Q_r^-(\Lambda|\Lambda^{(n)}) = -\frac{1}{2}\left(\mathbf{u}^T\Phi_r^-\mathbf{u} - 2\tilde{\mathbf{x}}_r'^T\Phi_r^-\mathbf{u} + \tilde{\mathbf{x}}_r'^T\Phi_r^-\tilde{\mathbf{x}}_r'\right) + b_r'' \tag{36}$$

where $\mathbf{u}$ is the same concatenated vector of normalized Gaussian means as in (28), Φ_r^- and $\tilde{\mathbf{x}}_r'$ are defined in the same way as their counterparts in (28) but computed with denominator statistics in (33) and (34) for utterance X_r.

Substituting (28) and (36) into (15), we can rewrite (15) to the final form for each support token, i.e.,

$$\mathbf{u}^T\mathbf{Q}_r\mathbf{u} + \mathbf{q}_r\mathbf{u} + g_r + 2\rho \le 0 \text{ for } X_r \in \mathcal{S} \tag{37}$$

where

$$\mathbf{Q}_r = \Phi_r^+ - \Phi_r^- \tag{38}$$

$$\mathbf{q}_r = 2\left(\tilde{\mathbf{x}}_r'^T\Phi_r^- - \tilde{\mathbf{x}}_r^T\Phi_r^+\right) \tag{39}$$

and

$$g_r = \tilde{\mathbf{x}}_r^T\Phi_r^+\tilde{\mathbf{x}}_r - \tilde{\mathbf{x}}_r'^T\Phi_r^-\tilde{\mathbf{x}}_r + 2b_r'' - 2b_r'. \tag{40}$$

Similarly, we can obtain the final constraint for error tokens in (16)

$$\mathbf{u}^T\bar{\mathbf{Q}}\mathbf{u} + \bar{\mathbf{q}}\mathbf{u} + \bar{g} - 2\rho' \le 0. \tag{41}$$

where

$$\bar{\mathbf{Q}} = \sum_{X_r\in\mathcal{E}}\left(\Phi_r^+ - \Phi_r^-\right) \tag{42}$$

$$\bar{\mathbf{q}} = 2\sum_{X_r\in\mathcal{E}}\left(\tilde{\mathbf{x}}_r'^T\Phi_r^- - \tilde{\mathbf{x}}_r^T\Phi_r^+\right) \tag{43}$$

and

$$\bar{g} = \sum_{X_r\in\mathcal{E}}\left(\tilde{\mathbf{x}}_r^T\Phi_r^+\tilde{\mathbf{x}}_r - \tilde{\mathbf{x}}_r'^T\Phi_r^-\tilde{\mathbf{x}}_r + 2b_r'' - 2b_r'\right). \tag{44}$$

Furthermore, the locality constraint in (17) can also be represented as a spherical constraint of $\mathbf{u}$ as follows:

$$\begin{aligned}\|\Lambda - \Lambda^{(n)}\| &= \sum_{k\in\mathcal{K}}\sum_{d=1}^{D}\frac{\left(\mu_{kd} - \mu_{kd}^{(n)}\right)^2}{\sigma_{kd}^2} \\ &= \|\mathbf{u} - \mathbf{u}^{(n)}\| \le h.\end{aligned} \tag{45}$$

After substituting the matrix-based formulations for both support and error tokens, *Problem* 1 can be converted into a new optimization problem as follows.

1) Problem 2:

$$\min_{\mathbf{u},\rho,\rho'}\left[-\rho + \epsilon\cdot\frac{1}{|\mathcal{E}|}\cdot\rho'\right] \tag{46}$$

subject to: $\mathbf{u}^T\mathbf{Q}_r\mathbf{u} + \mathbf{q}_r\mathbf{u} + g_r + 2\rho \le 0$ for all $X_r \in \mathcal{S}$ (47)

$$\mathbf{u}^T\bar{\mathbf{Q}}\mathbf{u} + \bar{\mathbf{q}}\mathbf{u} + \bar{g} - 2\rho' \le 0 \tag{48}$$

$$\|\mathbf{u} - \mathbf{u}^{(n)}\| \le h \tag{49}$$

$$\rho \ge 0 \text{ and } \rho' \ge 0. \tag{50}$$

From *Problem* 2, we can see that the constraint of (49) is convex. The constraints in (47) and (48) are in standard quadratic form, but they are not convex since we cannot guarantee matrices $\mathbf{Q}_r$ and $\bar{\mathbf{Q}}$ for support and error tokens are positive semi-definite. Therefore, the current form of the optimization problem of *Problem* 2 cannot be directly solved by the SOCP method. To resolve this problem, we have to resort some SOCP relaxation techniques. In this paper, we study three different SOCP relaxation methods: named as *RLX0, RLX1* and *RLX2*. We refer to the first one as the basic relaxation technique since it was firstly proposed by Kim *et al.* in [15]. The second and third ones are originally proposed in this work, which have been found to lead to much tighter relaxation and in turn yield much better estimation accuracy than the basic relaxation method for LME of HMMs.

A. Basic Relaxation Method: RLX0

In order to convert *Problem* 2 into a standard SOCP problem as in (19), some convex relaxation techniques have to be used. As the first attempt, we adopt the same SOCP relaxation method in [15], which is referred to as *RLX0* in this work, to convert matrices of $\mathbf{Q}_r$ and $\bar{\mathbf{Q}}$ into positive semi-definite matrices combined with a sequence of linear constraints. The basic idea in [15] is to decompose an indefinite matrix according to its eigenvectors and eigenvalues. More specifically, suppose $\lambda_r^1, \lambda_r^2, \ldots, \lambda_r^{\mathcal{M}}$ be all eigenvalues ($\mathcal{M} = \mathcal{K}D$ in total) of support token matrix $\mathbf{Q}_r$, where λ_r^k corresponds to the kth element of the diagonal matrix $\mathbf{Q}_r$, and let $\mathbf{v}_r^1, \mathbf{v}_r^2, \ldots, \mathbf{v}_r^{\mathcal{M}}$ be all $\mathcal{M}$ eigenvectors of $\mathbf{Q}_r$ corresponding to the eigenvalues $\lambda_r^1, \lambda_r^2, \ldots, \lambda_r^{\mathcal{M}}$ and satisfying $\left\|\mathbf{v}_r^k\right\| = 1$ and $\left(\mathbf{v}_r^m\right)^T\mathbf{v}_r^k = 0$, $k \ne m$. In case matrix $\mathbf{Q}_r$ is diagonal, they are just Euclidean base vectors. Then, it is easy to show that matrix $\mathbf{Q}_r$ can be decomposed into two parts as follows:

$$\mathbf{Q}_r = \mathbf{Q}_r^+ + \sum_{k\in\mathcal{M},\lambda_r^k<0}\lambda_r^k\cdot\mathbf{v}_r^k\left(\mathbf{v}_r^k\right)^T \tag{51}$$

where $\mathbf{Q}_r^+$ is constructed based on only eigenvectors corresponding to positive eigenvalues, as $\mathbf{Q}_r^+ = \sum_{k\in\mathcal{M},\lambda_r^k>0}\lambda_r^k\cdot\mathbf{v}_r^k\left(\mathbf{v}_r^k\right)^T$. Obviously, $\mathbf{Q}_r^+$ is a positive semi-definite matrix.

Substituting (51) into (47) and introducing a new variable z_r^k for each eigenvector with negative eigenvalue, in this way, we can derive the following two constraints which are equivalent to (47)

$$\mathbf{u}^T\mathbf{Q}_r^+\mathbf{u} + \sum_{k\in\mathcal{M},\lambda_r^k<0}\lambda_r^k z_r^k + \mathbf{q}_r\mathbf{u} + g_r + 2\rho \le 0 \tag{52}$$

with

$$z_r^k = \mathbf{u}^T\mathbf{v}_r^k\left(\mathbf{v}_r^k\right)^T\mathbf{u} = \tilde{\mu}_k^2, \forall\, k \in \mathcal{K} \text{ and } \lambda_r^k < 0. \tag{53}$$

where $\tilde{\mu}_k$ is a component in the normalized mean vector $\mathbf{u}$ in (29).

Obviously the constraint in (52) is convex quadratic, however, the nonlinear equality constraint in (53) still remains non-convex. Following [15], this equality constraint is relaxed

into an inequality constraint but upper-bounded by a constant as follows:

$$\tilde{\mu}_k^2 \le z_r^k \le C_k \forall\, k \in \mathcal{K} \text{ and } \lambda_r^k < 0 \tag{54}$$

where C_k is a constant. In our case, C_k can be roughly estimated from the locality constraint of (49) as $C_k = \left(\left|\tilde{\mu}_k^{(n)}\right| + h\right)^2$. In this paper, the above relaxation is named as *RLX0*.

Analogously, the error token matrix $\bar{\mathbf{Q}}$ can also be converted into a positive definite matrix $\bar{\mathbf{Q}}^+$ and a linear term consisting of all negative eigenvalues, i.e.,

$$\bar{\mathbf{Q}} = \bar{\mathbf{Q}}^+ + \sum_{m \in \mathcal{M}, \lambda^m < 0} \lambda^m \cdot \mathbf{v}^m (\mathbf{v}^m)^T \tag{55}$$

where $\bar{\mathbf{Q}}^+$ is constructed from all positive eigenvalues, as $\bar{\mathbf{Q}}^+ = \sum_{m \in \mathcal{M}, \lambda^m > 0} \lambda^m \cdot \mathbf{v}^m (\mathbf{v}^m)^T$. Obviously, $\bar{\mathbf{Q}}^+$ is positive semi-definite.

Using the same relaxation method as above, we can convert (48) into the following two convex constraints:

$$\mathbf{u}^T \bar{\mathbf{Q}}^+ \mathbf{u} + \sum_{m \in \mathcal{M}, \lambda^m < 0} \lambda^m z^m + \mathbf{q}\mathbf{u} + g + 2\rho \le 0 \tag{56}$$

with

$$\tilde{\mu}_m^2 \le z^m \le C_m \forall\, m \in \mathcal{M} \text{ and } \lambda^m < 0 \tag{57}$$

where C_m is a constant. Similarly, C_m can be also roughly estimated from the locality constraint of (49) as $C_m = \left(\left|\tilde{\mu}_m^{(n)}\right| + h\right)^2$.

Using the SOCP relaxation *RLX0*, *Problem* 2 can be converted into the following convex optimization problem.

1) Problem 3:

$$\min_{\mathbf{u},\rho,\rho'} \left[-\rho + \epsilon \cdot \frac{1}{|\mathcal{E}|} \cdot \rho'\right] \tag{58}$$

$$\text{subject to:} \quad \mathbf{u}^T \mathbf{Q}_r^+ \mathbf{u} + \sum_{k \in \mathcal{M}, \lambda_r^k < 0} \lambda_r^k z_r^k + \mathbf{q}_r \mathbf{u} + g_r + 2\rho \le 0 \text{ for all } X_r \in \mathcal{S} \tag{59}$$

$$\tilde{\mu}_k^2 \le z_r^k \le \left(\left|\tilde{\mu}_k^{(n)}\right| + h\right)^2 \forall\, k \in \mathcal{K} \text{ and } \lambda_r^k < 0 \tag{60}$$

$$\mathbf{u}^T \bar{\mathbf{Q}}^+ \mathbf{u} + \sum_{m \in \mathcal{M}, \lambda^m < 0} \lambda^m z^m + \mathbf{q}\mathbf{u} + g + 2\rho \le 0 \tag{61}$$

$$\tilde{\mu}_m^2 \le z^m \le \left(\left|\tilde{\mu}_m^{(n)}\right| + h\right)^2 \forall\, m \in \mathcal{M} \text{ and } \lambda^m < 0 \tag{62}$$

$$\|\mathbf{u} - \mathbf{u}^{(n)}\| \le h \tag{63}$$

$$\rho \ge 0 \text{ and } \rho' \ge 0. \tag{64}$$

The relaxation technique of *RLX0* can be intuitively illustrated in Fig. 1. The original LME problem can be viewed as optimizing an objective function along the solid curve segment in Fig. 1(a). After relaxation, optimization is actually conducted within the shaded area under the constant upper bound, which becomes a convex set.

However, as shown in Fig. 1(a), the relaxation of *RLX0* is taken quite loosely, due to a relatively large area of the relaxed region. Although it is helpful to seek a global solution, it inevitably introduces significant mismatch from the original optimization problem of LME. Because of this, we will consider two tighter relaxation methods to alleviate this problem in Sections IV-B and IV-C.

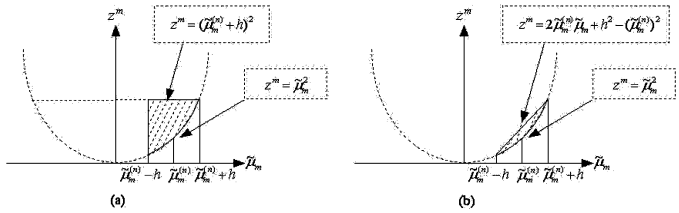

Fig. 1. Illustration of SOCP relaxation (RLX0 versus RLX1).

B. Mean-Dependent Linear Upper Bound on z^m: RLX1

The first method is to use a linear upper bound to replace a constant upper bound for z^m, and the relaxed convex area used for optimization thus becomes a sharp shaded area bounded by a line segment intersected with the original function curve, as plotted in Fig. 1(b). More specifically, the new relaxed constraint is upper-bounded by a linear function of z^m, which can be derived according to the locality constraint in (63)

$$\tilde{\mu}_m^2 \le z^m \le 2\tilde{\mu}_m^{(n)} \tilde{\mu}_m + h^2 - \left(\tilde{\mu}_m^{(n)}\right)^2. \tag{65}$$

This relaxation method is called *RLX1* in this paper. Obviously, the new relaxation area is still a convex set, but much smaller than *RLX0*.

Hence, using the relaxation method of *RLX1*, the LME optimization problem in *Problem* 2 can be reformulated as follows.

1) Problem 4:

$$\min_{\mathbf{u},\rho,\rho'} \left[-\rho + \epsilon \cdot \frac{1}{|\mathcal{E}|} \cdot \rho'\right] \tag{66}$$

$$\text{subject to:} \quad \mathbf{u}^T Q_r^+ \mathbf{u} + \sum_{k \in \mathcal{M}, \lambda_r^k < 0} \lambda_r^k z_r^k + \mathbf{q}_r \mathbf{u} + g_r + 2\rho \le 0 \tag{67}$$

$$\tilde{\mu}_k^2 \le z_r^k \le 2\tilde{\mu}_k^{(n)} \tilde{\mu}_k + h^2 - \left(\tilde{\mu}_k^{(n)}\right)^2 \text{ for all } k \in \mathcal{M} \text{ and } \lambda_r^k < 0 \text{ and } X_r \in \mathcal{S}. \tag{68}$$

$$\mathbf{u}^T \bar{Q}^+ \mathbf{u} + \sum_{m \in \mathcal{M}, \lambda_r^m < 0} \lambda_r^m z^m + \bar{\mathbf{q}}\mathbf{u} + \bar{g} - 2\rho' \le 0 \tag{69}$$

$$\tilde{\mu}_m^2 \le z^m \le 2\tilde{\mu}_m^{(n)} \tilde{\mu}_m + h^2 - \left(\tilde{\mu}_m^{(n)}\right)^2 \text{ for all } m \in \mathcal{M} \text{ and } \lambda^m < 0. \tag{70}$$

$$\|\mathbf{u} - \mathbf{u}^{(n)}\| \le h \tag{71}$$

$$\rho \ge 0 \text{ and } \rho' \ge 0. \tag{72}$$

C. Mean Shifting for Even Tighter SOCP Relaxation: RLX2

However, one potential problem for the relaxation method *RLX1* is that the relaxed area may not be ideally tight if the upper and lower bounds of $\tilde{\mu}_m$ have different signs, as shown in Fig. 2(a). The issue can be solved by a slightly improved relaxation method, which we refer to as *RLX2*. In this method, all the Gaussian means, i.e., $\tilde{\mu}_m$, are shifted by a constant positive value, e.g., d_m, towards right as $\tilde{\mu}'_m = \tilde{\mu}_m + d_m$ until the shaded relaxation area does not cross the origin for all Gaussian means. It is straightforward to show that the proposed constant shifting

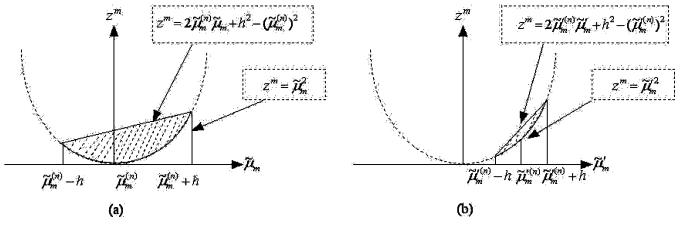

Fig. 2. Illustration of mean shift for tighter relaxation RLX2.

cannot reduce the area of the relaxed region but it can make the region longer and narrower so that it can significantly reduce the gap between the linear upper boundary and the underlying function.[1] Since d_m are all constant, they do not affect the optimization problem as long as we properly modify the objective function and all constraints based on the constant shifts. We first perform optimization with respect to the shifted mean $\tilde{\mu}'_m$ and the optimal solution of $\tilde{\mu}'_m$ is shifted back by the same constant to obtain the optimal value of the original Gaussian mean $\tilde{\mu}_m$. More importantly, if d_m is big enough, we can guarantee that the upper bound and lower bound of all shifted means $\tilde{\mu}'_m$ will have the same sign. Thus, the linear upper bound of z^m for both support and error token matrices in terms of $\tilde{\mu}'_m$ will achieve better approximation accuracy, as shown in Fig. 2(b). In our experiments, the value of d_m is determined based on Gaussian mean vectors in the initial models. In our experiments, a large enough value $d_m = 10$ is chosen to ensure the lower bounds of all shifted mean vector are all positive.

By using the relaxation method of *RLX2*, *Problem* 3 can be further relaxed and converted into the following convex optimization problem as follows.

1) Problem 5:

$$\min_{\mathbf{u}',\rho,\rho'} \left[-\rho + \epsilon \cdot \frac{1}{|\mathcal{E}|} \cdot \rho'\right] \tag{73}$$

$$\text{subject to:} \quad \mathbf{u}'^T \mathbf{Q}_r^+ \mathbf{u}' + \sum_{k\in\mathcal{M},\lambda_r^k<0} \lambda_r^k z_r^k + \left(\mathbf{q}_r - 2\mathbf{d}^T\mathbf{Q}_r^+\right)\mathbf{u}' + g_r + \mathbf{d}^T\mathbf{Q}_r^+\mathbf{d} - \mathbf{q}_r\mathbf{d} + 2\rho \le 0 \tag{74}$$

$$\tilde{\mu}_k'^2 \le z_r^k \le 2\tilde{\mu}_k'^{(n)}\tilde{\mu}_k' + h^2 - \left(\tilde{\mu}_k'^{(n)}\right)^2 \tag{75}$$

$$\text{for all } k \in \mathcal{M}, \lambda_r^k < 0 \text{ and } X_r \in \mathcal{S}. \tag{76}$$

$$\mathbf{u}'^T\bar{\mathbf{Q}}^+\mathbf{u}' + \sum_{m\in\mathcal{M},\lambda_r^m<0} \lambda_r^m z^m + (\bar{\mathbf{q}} - 2\mathbf{d}^T\bar{\mathbf{Q}}^+)\mathbf{u}' + \bar{g} + \mathbf{d}^T\bar{\mathbf{Q}}^+\mathbf{d} - \bar{\mathbf{q}}\mathbf{d} - 2\rho' \le 0 \tag{77}$$

$$\tilde{\mu}_m'^2 \le z^m \le 2\tilde{\mu}_m'^{(n)}\tilde{\mu}_m' + h^2 - \left(\tilde{\mu}_m'^{(n)}\right)^2 \tag{78}$$

$$\text{for all } m \in \mathcal{M} \text{ and } \lambda^m < 0. \tag{79}$$

$$||\mathbf{u}' - \mathbf{u}'^{(n)}|| \le h \tag{80}$$

$$\rho \ge 0, \rho' \ge 0 \text{ and } \mathbf{d} = [d_1, \ldots, d_{\mathcal{M}}]^T. \tag{81}$$

[1]In fact, it is this gap not the total area that determines relaxation errors in all the three SOCP methods, including RLX0, RLX1 and RLX2. For any of these three SCOP problems, it is easy to show that given constant $\mathbf{u}$, the larger z variables are, the smaller the value of the objective function becomes. This can be easily justified in (67) since any increase of z values can be compromised by reducing ρ value, assuming all $\mathbf{u}$ are constant. Because of this, the optimal solution to Problems 3, 4, and 5 is always located in the upper boundary of the relaxed region, never in the middle of the region.

In summary, the above three optimization problems, namely *Problem* 3, *Problem* 4, and *Problem* 5, are all standard convex quadratic programming problem and they all can be easily converted into SOCP problem as in [18] and [30] so that they can be solved by any SOCP solver.

V. Experiments

The experiments in this section are organized as two parts. In the first part, we carry out experiments on a small scale ASR task, i.e., the TIDIGITS connected digit string recognition task. The purposes of the experiments on the TIDIGITS task are two-fold. First, we evaluate the proposed LME/SOCP methods and compare them with other DT methods, such as MCE, as well as LME using other optimization methods, such as gradient descent (GD) and SDP. Second, we evaluate and compare three different relaxation methods and choose the most efficient one to further apply it to LVCSR tasks. In the second part, we conduct experiments to examine effectiveness of the LME/SOCP method on an LVCSR task using the WSJ-5k corpus.

A. Experiments on TIDIGITS

The proposed SOCP-based optimization methods for LME have been evaluated on the TIDIGITS database for connected digit string speech recognition in string level. In our experiments, only adult portion of the TIDIGITS corpus is used. The training set has 8623 digit strings (from 112 speakers) and the test set has 8700 strings (from other 113 speakers). Our model set consists of 11 whole-word CDHMMs representing all digits. Each HMM has 12 states and uses a simple left-to-right topology without state-skipping. Acoustic feature vectors consist of 39 dimensions (12 MFCCs and the normalized energy, plus their first- and second-order time derivatives). Different number of Gaussian mixture components (from 1 to 32 per state) are experimented. In all LME methods, we use the best MCE models (see [9]) as the initial models and only HMM mean vectors are re-estimated with LME. In each iteration of LME, a number of competing strings are generated for each utterance in training set based on its N-best decoding results ($N = 5$). Then we select support tokens according to (10) and obtain the optimal Viterbi sequence for each support token according to the recognition result. Then, in each iteration, the relaxed SOCP optimization is conducted with respect to $\mathbf{u}$, ρ, ρ' and z^m with the searching range h being set to 0.06. At last, CDHMM means are updated based on the optimization solution $\mathbf{u}^*$. In this paper, all convex optimization problems, i.e., *Problem* 3, *Problem* 4 and *Problem 5* are solved by an SOCP optimization tool, *MOSEK 4.0* [22], running under Matlab.

In our experiments, the SOCP method with three different types of relaxation techniques, namely *RLX0*, *RLX1*, and *RLX2*, have been compared with the LME using gradient descent (GD) method in [11], denoted as *GRAD*, and the SDP method in [16], denoted as *SDP*. We also include the ML and minimum classification error (MCE) [9] baseline systems in the table for reference. In Table I, we gives performance comparison on the TIDIGITS test set using all these different training methods. From the results, we can also see that the RLX0 method only achieves very little improvement over MCE method, while

TABLE I
STRING ERROR RATES (IN %) ON THE TIDIGITS TEST DATA

	ML	MCE	LME				
			GRAD	SDP	**RLX0**	**RLX1**	**RLX2**
1-mix	12.61	6.72	3.77	2.75	6.67	2.68	**2.10**
2-mix	5.26	3.94	1.70	1.24	3.85	1.23	**1.13**
4-mix	3.48	2.23	1.24	0.89	1.83	0.99	**0.98**
8-mix	1.94	1.41	0.87	0.68	1.30	0.80	**0.76**
16-mix	1.72	1.11	0.82	0.63	1.10	0.71	**0.66**
32-mix	1.34	0.90	0.66	0.53	0.90	0.66	**0.59**

TABLE II
AVERAGE OPTIMIZATION TIME (IN SECONDS) PER ITERATION

	1-mix	2-mix	4-mix	8-mix	16-mix	32-mix
SDP	1275.6	1068.5	3556.4	12398.0	40691.0	113110.0
RLX1	57.2	58.8	141.5	192.5	324.9	506.4
speedup	x22	x18	x25	x64	x125	x223

RLX1 and RLX2 method not only significantly improve recognition performance over the MCE method, but also largely outperform the simple GD-based LME method. By applying mean shifting, RLX2 achieves the best recognition performance among three SOCP approaches. Also all LME/SOCP methods (RLX0, RLX1, and RLX2) are relatively easy to run while the GD method needs lots of fine-tuning on its parameters such as step size and penalty weight coefficients. From the results, we can see that the LME/SDP method in [11] still achieves the best overall performance, especially for large models. However, the performance gap between LME/SDP and our best RLX2 is not significant. It is safe to say that the RLX2 yields comparable performance as LME/SDP on the TIDIGITS task. However, if we compare the efficiency between LME/SDP and our proposed SOCP approaches, all three SOCP methods[2] run substantially faster and consume much less memory during optimization. As one example, the CPU times needed to optimize each problem per iteration are listed in Table II for comparison between LME/SDP and RLX1. It is clear that the RLX1 method runs about 20–200 times faster than the SDP method. The speed gap between them grows bigger when the model size increases. This can be easily explained because an SOCP problem can be solved more efficiently than an SDP problem with the similar problem size and structure. Moreover, the size of optimization variable matrix in the LME/SDP [11] is proportional to square of total number of Gaussians while the size of optimization variable (e.g., $\mathbf{u}$) is only proportional to the number of Gaussians. As a result, for the same CDHMM model set, the problem size of SOCP is significantly smaller than the LME/SDP.

The advantage of SOCP with faster training speed and less memory consumption is critical if we extend the LME method to large-scale ASR tasks, such as LVCSR, which typically involves much larger HMM model sets. The requirements of a large amount of memory and computing resources will probably make it difficult to apply the SDP method to any state-of-the-art LVCSR task. In Section V-B, we will present our experiments in an LVCSR task to further evaluate effectiveness of the proposed LME/SOCP method on some larger scale ASR tasks. However, based on the experiments on TIDIGITS, we only focus on the best relaxation method, i.e., RLX2, due to its proved efficiency and effectiveness, while the other two relaxation methods are not explored furthermore.

[2]Experimental results show all three SOCP methods need similar optimization time and memory.

B. Experiments on WSJ-5k

In this section, the proposed LME/SOCP method with relaxation *RLX2* is evaluated on a standard large-vocabulary continuous speech recognition task using the WSJ0 (Wall Street Journal) corpus. In our experiments, we use the standard SI-84 set as training data, which consists of 7133 training utterances from 84 speakers. Evaluation has been performed on the standard Nov'92 nonverbalized 5 k close-vocabulary test set (WSJ-5k), including 330 utterances in total from other eight speakers. Feature extraction uses the standard 39 dimensional vector, including 12 Mel-frequency cepstral coefficients (MFCCs) and normalized energy, along with their delta and acceleration coefficients. In our baseline model, we use the HTK tools [31] to build a crossword tri-phone HMMs with a total number of 2774 tied-states and each tied state has eight Gaussian components. A standard trigram language model has been used for test. The maximum-likelihood estimation (MLE) baseline system achieves 95.35% in word accuracy, which is comparable with other systems reported in the same task.

We use the above MLE models as the seed model to conduct discriminative training based on word graphs. The word graphs are generated for all training data using the MLE models based on a uni-gram LM trained from all training transcriptions. In these experiments, the proposed LME/SOCP method has been compared with a different DT method, i.e., maximum mutual information estimation (MMIE) and boosted MMIE using the popular EBW optimization method, which is implemented using the provided HTK tool [31]. All the parameters related to the MMIE/EBW method are set to the default values suggested by HTK. In the MMIE/EBW method, the learning constant E was set to 2, i-smoothing $\tau = 100$, acoustic scale factor $k = 1/15$. In the LME method, *Problem* 5 is solved by the commercial SOCP solver, i.e., MOSEK 4.0 [22], running under Matlab. As in [27], the value for searching range h in SOCP optimization procedure is set to 0.005.

As shown in (74) and (75) of Problem 5, we need to introduce a set of constraints for every sentence in the selected support token set $\mathcal{S}$. As the support token set $\mathcal{S}$ grows, the number of total constraints in the optimization Problem 5 also increases dramatically. This will inevitably increase complexity of the optimization problem and it requires more CPU times and memory to solve the underlying SOCP problem. Therefore, we need to limit the total number of constraints in the optimization problem for better efficiency in the training process. In this paper, we split all support tokens in $\mathcal{S}$ based on their margins into a smaller number of groups. In this way, instead of introducing one set of constraints in (74) and (75) for every token in $\mathcal{S}$, we only need to impose a set of constraints for each group based on the average margins of all tokens in this group.

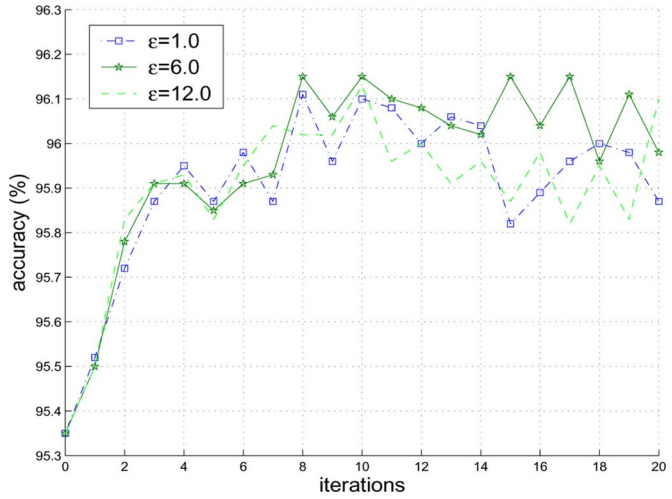

Fig. 3. Effect of various control parameter ϵ between support and error tokens on WSJ-5k test set.

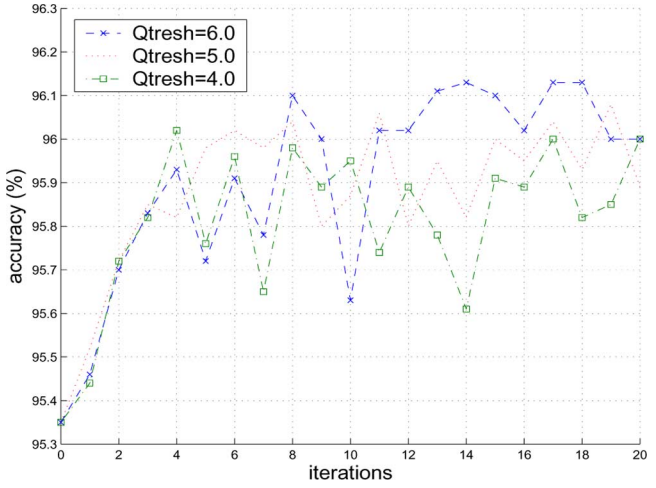

Fig. 4. Recognition performance on WSJ-5k test set by varying threshold $Qtresh$ to prune matrix $\mathbf{Q}$.

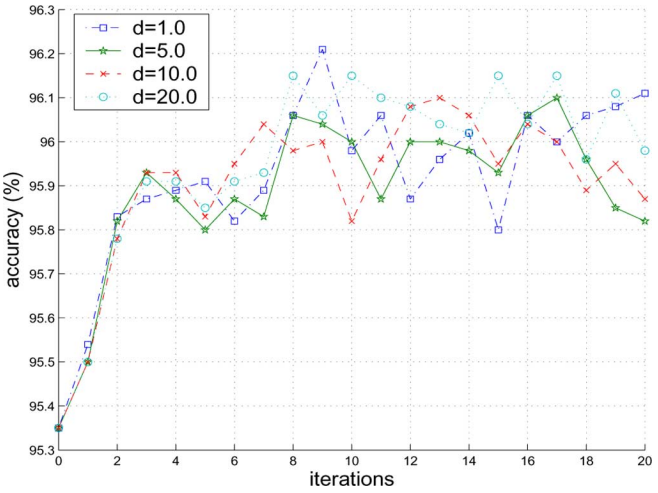

Fig. 5. Effect of support token selection parameter d on WSJ-5k test set.

In the following, we conduct three different sets of ASR experiments on the WSJ-5k task to investigate the performance of the LME/SOCP method. First of all, we investigate the effect of the contribution between support and error tokens by varying the value of ϵ; Second, we study the effect of using different number of support tokens in the LME/SOCP optimization where all support tokens are assumed to belong to one group; Third, we study the effect of using different number of groups to partition the support token set. Finally, we will compare the best performance of the proposed LME/SOCP method with other conventional DT training methods, such as MMIE and boosted MMIE using the EBW method.

1) Effect of the Contribution Between Support and Error Tokens (ϵ)*:* In this experiment, we investigate the effect of the contribution between support and error tokens to the final system performance by varying the interpolation parameter ϵ. The values we used include 1.0, 6.0, and 12.0. In Fig. 3, we can see that the best performance was obtained by setting the value of ϵ to 6.0. Therefore, for other experiments, we set 6.0 to the parameter ϵ.

2) Effect of Pruning Small Values of the Matrix Q*:* As shown in [27], small values in the matrix Q introduce noise into an optimization procedure and are therefore harmful for system training. A pruning strategy with the threshold parameter $Qthresh$ has to be used to remove the small values (posterior probabilities) from the matrix Q. To this end, we varied the value of $Qthresh$ by 6.0, 5.0, and 4.0 but kept all other parameters unchanged. It was found that $Qthresh = 6.0$, which kept approximately 25% elements of the matrix Q, achieved the best performance as shown in Fig. 4. With this type of pruning, the number of parameters that are optimized in SOCP is reduced to 25% and the remaining 75% of Gaussian mean values are not re-estimated during each iteration. For other experiments thereafter, the $Qthresh$ is set to 6.0.

3) Effect of Support Token Selection: In this experiment, we check the impact from the selection of a different number of support tokens. We fix all the other parameters unchanged but use a different threshold, i.e., d in (10), to select support token sets from all training data. Here, we consider four different values for d, i.e., d =1.0, 5.0, 10.0, 20.0. In this experiment, we only use one group for all selected support tokens.

From Fig. 5, it is clearly shown that the number of support tokens strongly affects the final recognition performance. Among them, the threshold value $d = 1.0$ yields the best recognition performance. As a result, we will use $d = 1.0$ for the other experiments thereafter.

4) Effect of Different Partition Groups: In this experiment, we examine the effect of different group numbers for the support token set on the final system performance. The best setting $d = 1.0$ is used to select support tokens and all selected support tokens are equally split into a number of groups according to their margins.

In the experiment shown in Fig. 6, we consider to split support tokens into two, three and four different groups. From Fig. 6, we can see that the results with three group partition achieve the best performance.

5) Performance Comparison With MLE, MMIE and Boosted-MMIE: In this experiment, we compare the performance of LME with the conventional MLE training method and two other DT training methods based on MMIE and boosted MMIE criteria using the standard EBW method. The parameter setting we

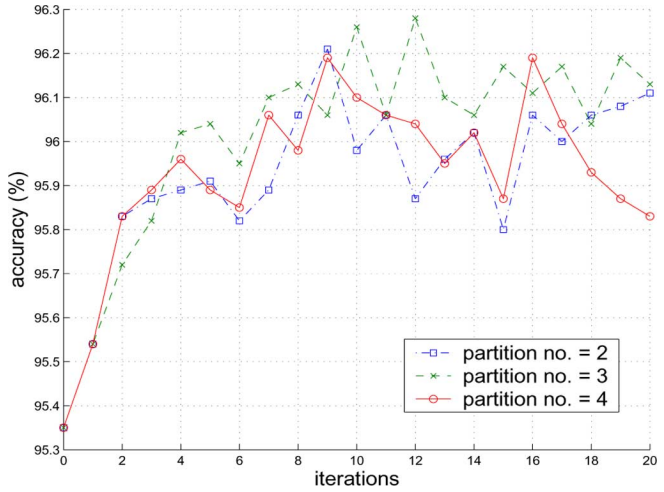

Fig. 6. Recognition performance of different partitioning strategies on WSJ-5k test set.

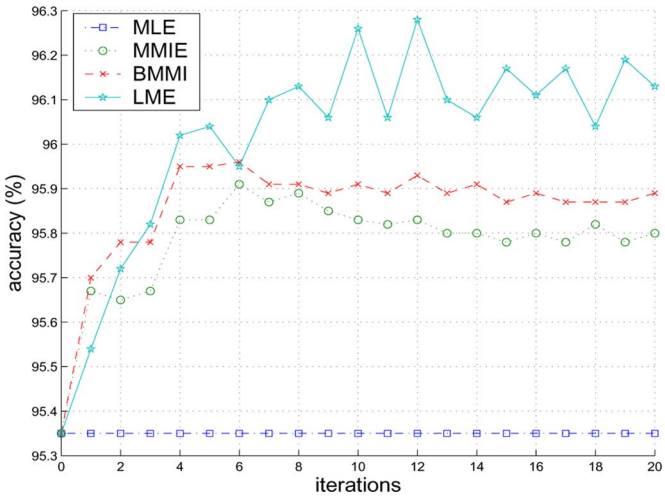

Fig. 7. Comparison of MLE, MMIE, Boosted MMIE, and LME/SOCP. Note: LME-SOCP only updates the mean vectors of the model, whereas the other three methods update all the model parameters, including weights, means, and variances.

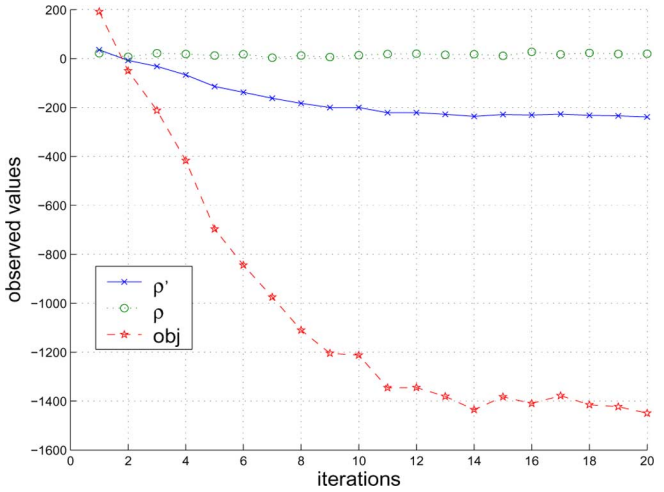

Fig. 8. Parameters ρ, ρ' and the objective function evolve during LME/SOCP training process.

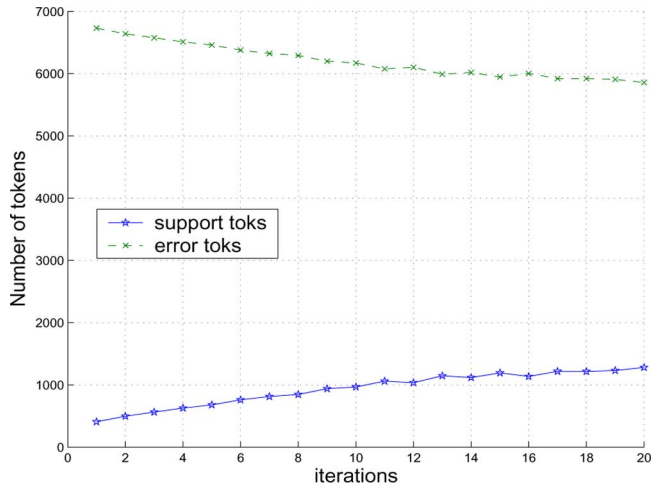

Fig. 9. Number of support and error tokens evolve during LME/SOCP training process.

used in this experiment is $d = 1.0$, partition no. $= 3$. The algorithm of the boosted MMIE is implemented according to the description of [25] by subtracting from the acoustic log-likelihood the boosting parameter times the contribution of the sentence-level accuracy arising from each arc in the forward–backward algorithm [25]. The boosting parameter of the boosted MMIE is set to 0.5, which has been tuned in a range $(0, 2.0]$ for the best performance according to [25]. All the other parameters of the boosted MMIE are set to the same values used by MMIE. In Fig. 7, we can see that the performance of the boosted MMIE converges more quickly than that of the conventional MMIE. The performance of the proposed LME/SOCP method with updating only mean vectors improves the result of the MMIE baseline result with updating all the model parameters, including weights, means and variances, from $\mathbf{95.91}\%$ to $\mathbf{96.28}\%$, with relative 9% of error reduction. We also carried out significance test, which showed the LME-SOCP method significantly outperforms MMIE-based DT training method (the p-value 0.017 is obtained at the level of 0.05 with the matched pair sentence segment test using the standard NIST significance test package for automatic speech recognition, namely sclite.[3]

We also present some representative evolution curves regarding the margin parameters ρ, ρ' and the number of support tokens during the iterations of optimization, which is helpful to demonstrate the essence of the equations in Problem 5. The curves of the above parameters are produced based on the optimal settings for the LME/SOCP method, i.e., $d = 1.0$, partition no. $= 3$, $Qthresh = 6.0$ and $\epsilon = 6.0$. It is clearly shown in Fig. 8 that the objective function decreases with iterations in the optimization process, which demonstrates the effective of the optimization procedure as shown in Problem 5. Besides this, we also demonstrate the evolution trend of the numbers of support and error tokens with optimization iterations in Fig. 9.

By interpreting the experimental results between MLE, MMIE/boosted MMIE and LME/SOCP, we can see that the improvement of LME/SOCP mainly comes from three parts.

[3]available at http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm.

First, all the MMIE/Boosted MMIE and LME/SOCP significantly outperform the conventional MLE training method because they have all utilized DT information in their training procedures. This part of the improvement is mostly due to the advantages of DT training over GT training. Second, some additional improvement for LME comes from a better generalization capacity held by LME than the other DT criteria, e.g., MMIE. Third, another part of the improvement may be attributed to the application of a more effective convex optimization method. As it is shown in the experiments, the same LME criterion gives quite different performance when different optimization algorithms, such as GD, SDP, SOCP, are used (cf. Table I). Finally, the LME/SOCP can guarantee to find a global solution, though some relaxation is applied, but the conventional EBW method is suboptimal so that it may get stuck in a local optimal solution.

VI. Conclusion

In this paper, we have proposed to use SOCP for LME of CDHMMs in speech recognition. The two new SOCP relaxation methods, namely RLX1 and RLX2, have been proposed to convert LME of Gaussian mixture HMMs into an SOCP problem and they both have been demonstrated to be effective in several speech recognition tasks. The proposed formulation is general enough to deal with various types of competing hypothesis space, such as N-best lists and word graphs. Our experimental results have shown that the proposed LME/SOCP method has achieved better performance than other DT methods in a small-scale connected digit string recognition task using TIDIGITS database as well as a large-vocabulary continuous speech recognition task using the WSJ0 corpus.

References

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

[2] 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.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[4] W. Chou, B.-H. Juang, and C.-H. Lee, "Segmental GPD training of HMM based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 1992, vol. 1, pp. 473–476.

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

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. R. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.

[7] G. Heigold, T. Deselaers, R. Schlueter, and H. Ney, "Modified MMI/MPE: A Direct evaluation of the margin in speech recognition," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Helsinki, Finland, 2008.

[8] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.

[9] 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.

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

[12] H. Jiang and X. Li, "Incorporating training errors for large margin HMMs under semi-definite programming framework," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Hawaii, USA, 2007, pp. 629–632.

[13] H. Jiang, "Discriminative training for automatic speech recognition: A survey," *Comput. Speech Lang.*, vol. 24, no. 4, pp. 589–608, Oct. 2010.

[14] H. Jiang and X. Li, "Parameter estimation of statistical models using convex optimization: An advanced method of discriminative training for speech and language processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 115–127, May 2010.

[15] S. Kim and M. Kojima, "Second order cone programming relaxation of non-convex quadratic optimization problems," *Optimization Methods and Software*, vol. 15, no. 3–4, pp. 201–224, 2001.

[16] X. Li and H. Jiang, "Solving large margin hidden Markov model estimation via semidefinite programming," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 8, pp. 2383–2392, Nov. 2007.

[17] P. Liu, C. Liu, H. Jiang, F. Soong, and R.-H. Wang, "A constrained line search optimization method for discriminative training of HMMs," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 5, pp. 900–909, Jul. 2008.

[18] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second order cone programming," *Linear Algebra Applicat.*, no. 284, pp. 193–228, 1998.

[19] W. Macherey, L. Haferkamp, R. Schluter, and H. Ney, "Investigations on error minimizing training criteria for discriminative training in automatic speech recognition," in *Proc. Interspeech*, Lisbon, Portugal, 2005, pp. 2133–2136.

[20] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large-vocabulary speech recognition using minimum classification error," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 203–223, 2007.

[21] E. McDermott, S. Watanabe, and A. Nakamura, "Margin-space integration of MPE loss via differencing of MMI functionals for generalized error-weighted discriminative training," *Proc. InterSpeech*, 2009.

[22] The Mosek Optimization Tools (User's Manual and Reference) Mosek ApS Inc., 2006 [Online]. Available: http://www.mosek.com

[23] 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)*, 2002, pp. 105–108.

[24] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Dept. Eng., Cambridge Univ., Cambridge, U.K., 2004.

[25] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 4057–4060.

[26] J. Weston and C. Watkins, "Support vector machines for multiclass pattern recognition," in *Proc. Eur. Symp. Artif. Neural Netw.*, 1999.

[27] D. Wu, B. J. Li, and H. Jiang, "Maximum mutual information estimation via second order cone programming for large vocabulary continuous speech recognition," *Proc. Interspeech*, Sep. 2009.

[28] Y. Yin and H. Jiang, "A fast optimization method for large margin estimation of HMMs based on second order cone programming," *Proc. Interspeech*, Sep. 2007.

[29] Y. Yin and H. Jiang, "A compact semidefinite programming (SDP) formulation for large margin estimation of HMMs in speech recognition," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding (ASRU)*, Kyoto, Japan, Dec. 2007, pp. 312–317.

[30] Y. Yin, "A Study of Convex Optimization for Discriminative Training of Hidden Markov Models in Automatic Speech Recognition," M.S. thesis, Dept. Comput. Sci. and Eng., York Univ., Toronto, ON, Canada, 2007.

[31] S. Young *et al., The HTK Book V3.2*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

Dalei Wu received the B.S. degree in computer science from Harbin Engineering University, Harbin, China, in 1996, the M.S. degree in computer science from Tsinghua University, Beijing, China in 2002, and the Ph.D. degree in computational linguistics from Saarland University, Saarbruecken, Germany, in 2006.

Since 2008, he has been a Postdoctoral Researcher in the Department of Computer Science and Engineering, York University, Toronto, ON, Canada. His research interest focuses on automatic speech recognition, automatic speaker recognition, and machine learning algorithms.

Yan Yin received the B.S. and M.S. degree from the Department of Computer Science and Engineering, York University, Toronto, ON, Canada, in 2007.

He is currently a Speech Research Engineer with Li Creative Technologies, Inc., Florham Park, NJ. He is working on advanced speech-related techniques for desktop and handheld devices including speech recognition, speaker recognition, and text-to-speech. His major research interests focus on automatic speech and speaker recognition, especially acoustic model 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 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 was with 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.

Dr. Jiang has served as an Associate Editor for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING since 2009. 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.