A Fast Optimization Method for Large Margin Estimation of HMMs Based on Second Order Cone Programming

Yan Yin, Hui Jiang

Department of Computer Science and Engineering, York University,
4700 Keele Street, Toronto, Ontario M3J 1P3, CANADA
Email: {yyin, hj}@cse.yorku.ca

10.21437/Interspeech.2007-6

ABSTRACT

In this paper, we present a new fast optimization method to solve large margin estimation (LME) of continuous density hidden Markov models (CDHMMs) for speech recognition based on *second order cone programming (SOCP)*. SOCP is a class of nonlinear convex optimization problems which can be solved quite efficiently. In this work, we have proposed a new convex relaxation condition under which LME of CDHMMs can be formulated as an SOCP problem. The new LME/SOCP method has been evaluated in a connected digit string recognition task using the TIDIGITS database. Experimental results clearly demonstrate that the LME using SOCP outperforms the previous gradient descent method and can achieve comparable performance as our previously proposed semidefinite programming (SDP) approach. But the SOCP yields much better efficiency in terms of optimization time (about 20-200 times faster) and memory usage when compared with the SDP method.
Index Terms: Large Margin Estimation (LME), CDHMM, Convex Optimization, Second Order Cone Programming (SOCP), Discriminative Training

1. INTRODUCTION

Recently, we have proposed large margin estimation (LME) of HMMs for speech recognition [1, 2, 4], 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, a large margin classifier implies a good generalization power and generally yields much lower generalization errors in new test data. As shown in [1], the estimation of large margin HMMs turns out to be a minimax optimization problem. In the past few years, several optimization methods have been proposed to solve this problem, such as *iterative localized optimization (ILO)* in [1], *constrained joint optimization (CJO)* method in [2], and *semidefinite programming (SDP) method* in [4]. However, there are still some limitations for all of these methods. The ILO method in [1] is hard to extend to continuous speech recognition tasks. In the CJO method [2], the derived unconstrained minimization problem is still a non-convex optimization problem which can only be solved with a general steepest gradient descent method. However, the gradient descent method can only lead to locally optimal solution which highly depends on the initial models and it is also hard to run experiments due to manual tuning of a number of sensitive parameters is needed. In [4], a new optimization method based on semidefinite programming (SDP) has been proposed for LME of CDHMMs in speech recognition. The SDP problem can be solved by many algorithms, such as [5] and many others, which lead to the globally optimal solution since SDP is a well-defined convex optimization. It is shown that the SDP method outperforms all other methods and it has achieved one of the best performances in the TIDIGITS task [4]. However, optimization time of the LME/SDP method increases dramatically as the size of HMM models grow because size of the SDP variable matrix in [4] (i.e., Z) is roughly equal to square of total number of Gaussians in the model set. In [4], we have successfully managed to handle a CDHMM set consisting of about 4k Gaussians but it is unlikely to directly extend the LME/SDP method in [4] to other larger scale speech recognition tasks which involve tens or even hundreds of thousands of Gaussians.

In this paper, we propose to use a different convex optimization method, namely *second order cone programming (SOCP)*, to solve the LME problem of CDHMMs in speech recognition. We will show that the LME problem can be converted into an SOCP problem under a different relaxation condition. In this LME/SOCP formulation, the size of SOCP variable vector is only proportional to the total number of Gaussians (not square as in LME/SDP). Moreover, 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 [3, 6, 7]. But just like SDP, an SOCP algorithm can also guarantee to find the globally optimal solution since SOCP is also a well-defined convex optimization problem. The proposed LME/SOCP method has been evaluated in a connected digit string recognition task using the TIDIGITS database. Experimental results show that the LME/SOCP method outperforms the previous gradient descent based methods and it can achieve comparable performance as the LME/SDP approach. But the LME/SOCP yields much better efficiency in terms of optimization time (about 20-200 times faster for various model sizes) and memory usage comparing with the previous LME/SDP method in [4].

2. LARGE MARGIN ESTIMATION OF HMMS

From [1, 2], the separation margin for a speech utterance X_i in a multi-class classifier is 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, λ_W denotes the HMM representing a word or word sequence W, W_i is the true word identity for X_i and $\mathcal{F}(X|\lambda_W)$ is called discriminant function, which is usually calculated in the logarithm scale: $\mathcal{F}(X|\lambda_W) =$

$\log[p(W)\cdot p(X|\lambda_W)]$. In this work, we are only interested in estimating HMMs λ_W and assume $p(W)$ is fixed.

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

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

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

The large margin principle leads to estimating the HMM models Λ based on the criterion of maximizing the minimum margin of all support tokens, which is named as large margin estimation (LME) of HMM.

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

3. SECOND ORDER CONE PROGRAMMING (SOCP)

A second order cone programming (SOCP) problem is a nonlinear convex optimization problem in which a linear function is minimized over the intersection of an affine set and the product of various second-order cones. A standard SOCP has the following form:

$$\begin{aligned} &\text{minimize} \quad f^T x \\ &\text{subject to} \quad \|A_i x + b_i\| \leq c_i^T x + d_i \quad (i = 1, \ldots, N) \end{aligned} \tag{4}$$

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., $\|u\| = (u^T u)^{\frac{1}{2}}$. We call the constraint $\|A_i x + b_i\| \leq c_i^T x + d_i$ in eq.(4) a 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. 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. Moreover, since SOCP is a well-defined convex optimization problem, the efficient algorithm for SOCP can lead to the globally optimal solution.

4. SECOND ORDER CONE PROGRAMMING RELAXATION

At first, we assume there are totally $\mathcal{K}$ Gaussian mixtures in the CDHMM set. For notational convenience, we denote them as $\mathcal{N}(\mu_k, \Sigma_k)$ where $k \in (1, 2, \ldots, \mathcal{K})$. Given any speech utterance $X_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{iT}\}$, as in [2], if we only estimate Gaussian mean vectors, the decision margin $d_{ij}(X_i)$ in eq.(1) can be represented as a standard diagonal quadratic form as:

$$\begin{aligned} d_{ij}(X_i) &= \mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_j) \\ &\approx c_{ij} - \frac{1}{2}\sum_{t=1}^{T}\sum_{d=1}^{D}\Big[\frac{(x_{itd} - \mu_{i_t d})^2}{\sigma_{i_t d}^2} - \frac{(x_{itd} - \mu_{j_t d})^2}{\sigma_{j_t d}^2}\Big] \end{aligned} \tag{5}$$

where we denote the optimal Viterbi path of X_i against λ_{W_i} as $\mathbf{i} = \{i_1, i_2, \cdots, i_T\}$, and the optimal Viterbi path against λ_j as $\mathbf{j} = \{j_1, j_2, \cdots, j_T\}$, and c_{ij} is a constant independent of all Gaussian means.

Since the margin as defined in eq.(5) is actually unbounded for CDHMMs, we adopt the following spherical constraint to guarantee the boundedness of margin as in [4]:

$$R(\Lambda) = \sum_{k=1}^{\mathcal{K}}\sum_{d=1}^{D}\frac{(\mu_{kd} - \mu_{kd}^{(0)})^2}{\sigma_{kd}^2} \leq r^2 \tag{6}$$

where r is a pre-set constant, and $\mu_{kd}^{(0)}$ is also a constant which is set to be the value of μ_{kd} in the initial models.

As shown in [4], the minimax optimization problem in eq.(3) becomes solvable under the constraint eq.(6). Following [4], we introduce a new variable $-\rho$ ($\rho > 0$) as the common upper bound for all terms in the minimax optimization, we can convert the minimax optimization in eq.(3) into an equivalent minimization problem as follows:

Problem 1

$$\Lambda^* = \arg\min_{\Lambda,\rho} -\rho \tag{7}$$

subject to

$$\mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) \leq -\rho \tag{8}$$

$$R(\Lambda) = \sum_{k=1}^{\mathcal{K}}\sum_{d=1}^{D}\frac{(\mu_{kd} - \mu_{kd}^{(0)})^2}{\sigma_{kd}^2} \leq r^2 \tag{9}$$

$$\rho \geq 0. \tag{10}$$

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

Now we consider how to solve the above LME of CDHMMs with the SOCP method. First of all, we introduce some notations: A column vector x is written as $x = (x_1; x_2; \ldots; x_n)$ and a row vector as $x = (x_1, x_2, \ldots, x_n)$. $\mathbf{I}_D$ is a $D \times D$ identity matrix. $\mathbf{0}_D$ is a $D \times D$ zero matrix. And $\mathbf{u}$ is a large column vector created by concatenating all normalized Gaussian mean vectors as:

$$\mathbf{u} = (\tilde{\mu}_1; \tilde{\mu}_2; \ldots; \tilde{\mu}_{\mathcal{K}}) \tag{11}$$

where each normalized Gaussian mean vector is

$$\tilde{\mu}_k := (\frac{\mu_{k1}}{\sigma_{k1}}; \frac{\mu_{k2}}{\sigma_{k2}}; \ldots; \frac{\mu_{kD}}{\sigma_{kD}}). \tag{12}$$

In the following, we will consider how to convert the minimization **Problem 1** into an SOCP problem as shown in eq.(4).

Firstly, we will formulate the constraint in eq.(9) into the standard second order cone constraint form shown in eq.(4):

$$\begin{aligned} R(\Lambda) &= \sum_{k=1}^{L}(\tilde{\mu}_k - \tilde{\mu}_k^{(0)})^T(\tilde{\mu}_k - \tilde{\mu}_k^{(0)}) \\ &= (\mathbf{u} - \mathbf{u}^{(0)})^T(\mathbf{u} - \mathbf{u}^{(0)}) = \|\mathbf{u} - \mathbf{u}^{(0)}\|^2 \leq r^2 \end{aligned} \tag{13}$$

where $\tilde{\mu}_k^{(0)}$ denotes one normalized initial Gaussian mean as in eq. (12).

Secondly, we will re-formulate the constraint in eq.(8) into the same standard second order cone constraint form. Suppose $\boldsymbol{\mu}_\mathbf{i}$ and $\boldsymbol{\mu}_\mathbf{j}$ denote two large column vectors created by concatenating all normalized Gaussian mean vectors along the Viterbi paths $\mathbf{i}$ and $\mathbf{j}$ respectively, i.e., $\boldsymbol{\mu}_\mathbf{i} = (\tilde{\mu}_{i_1}; \ldots; \tilde{\mu}_{i_T})$, $\boldsymbol{\mu}_\mathbf{j} = (\tilde{\mu}_{j_1}; \ldots; \tilde{\mu}_{j_T})$. And $\tilde{\mathbf{x}}_\mathbf{i}$ and $\tilde{\mathbf{x}}_\mathbf{j}$ denote two concatenated feature vectors in $X_i =$

$(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{iT})$ normalized by Gaussian variance vectors along the Viterbi paths $\mathbf{i}$ and $\mathbf{j}$ respectively, with $\tilde{\mathbf{x}}_\mathbf{i} = (\tilde{\mathbf{x}}_{i1}^i; \ldots; \tilde{\mathbf{x}}_{i_T}^i)$, $\tilde{\mathbf{x}}_\mathbf{j} = (\tilde{\mathbf{x}}_{i1}^j; \ldots; \tilde{\mathbf{x}}_{iT}^j)$, and $\tilde{\mathbf{x}}_{it}^i = (x_{it1}/\sigma_{i_t 1}; \ldots; x_{itD}/\sigma_{i_t D})$, $\tilde{\mathbf{x}}_{it}^j = (x_{it1}/\sigma_{j_t 1}; \ldots; x_{itD}/\sigma_{j_t D})$.

We construct a large $DR \times D\mathcal{K}$ matrix $\Phi_\mathbf{i}$ according to the above Viterbi path $\mathbf{i}$ as follows:

$$\Phi_\mathbf{i} = \begin{pmatrix} \overbrace{\mathbf{0}_D \quad \ldots \quad \mathbf{0}_D}^{i_1-1} \ \mathbf{I}_D \quad \ldots \mathbf{0}_D \ldots \\ \overbrace{\mathbf{0}_D \quad \ldots \quad \mathbf{0}_D}^{i_2-1} \ \mathbf{I}_D \quad \ldots \mathbf{0}_D \ldots \\ \vdots \qquad\qquad\qquad\quad \vdots \\ \overbrace{\mathbf{0}_D \quad \ldots \quad \mathbf{0}_D}^{i_R-1} \ \mathbf{I}_D \quad \ldots \mathbf{0}_D \ldots \end{pmatrix} \tag{14}$$

Obviously, $\mathbf{u}$ (in eq.(11)) and $\boldsymbol{\mu}_\mathbf{i}$ satisfy:

$$\boldsymbol{\mu}_\mathbf{i} = \Phi_\mathbf{i}\mathbf{u}. \tag{15}$$

Similarly, we have

$$\boldsymbol{\mu}_\mathbf{j} = \Phi_\mathbf{j}\mathbf{u}. \tag{16}$$

Therefore, we rewrite the eq.(5) as:

$$\begin{aligned} -d_{ij}(X) &= \mathcal{F}(X|\lambda_j) - F(X_i|\lambda_{W_i}) \\ &= c_{ji} - \left[\frac{1}{2}(\boldsymbol{\mu}_\mathbf{j} - \tilde{\mathbf{x}}_\mathbf{j})^T(\boldsymbol{\mu}_\mathbf{j} - \tilde{\mathbf{x}}_\mathbf{j}) - \frac{1}{2}(\boldsymbol{\mu}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{i})^T(\boldsymbol{\mu}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{i})\right] \\ &= c_{ji} - \frac{1}{2}\left[\mathbf{u}^T(\Phi_\mathbf{j}^T\Phi_\mathbf{j} - \Phi_\mathbf{i}^T\Phi_\mathbf{i})\mathbf{u} + 2(\tilde{\mathbf{x}}_\mathbf{i}^T\Phi_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}^T\Phi_\mathbf{j})\mathbf{u} + (\tilde{\mathbf{x}}_\mathbf{j}^T\tilde{\mathbf{x}}_\mathbf{j} - \tilde{\mathbf{x}}_\mathbf{i}^T\tilde{\mathbf{x}}_\mathbf{i})\right] \end{aligned} \tag{17}$$

If we denotes $Q_{ij} = \Phi_\mathbf{i}^T\Phi_\mathbf{i} - \Phi_\mathbf{j}^T\Phi_\mathbf{j}$, and $q_{ij} = 2(\tilde{\mathbf{x}}_\mathbf{j}^T\Phi_\mathbf{j} - \tilde{\mathbf{x}}_\mathbf{i}^T\Phi_\mathbf{i})$, and $g_{ij} = (\tilde{\mathbf{x}}_\mathbf{i}^T\tilde{\mathbf{x}}_\mathbf{i} - \tilde{\mathbf{x}}_\mathbf{j}^T\tilde{\mathbf{x}}_\mathbf{j}) + 2c_{ji}$. It is easy to show that Q_{ij} is a diagonal matrix. After applying all of these to eq.(17), we rewrite the constraint in eq.(8) as follows:

$$\mathbf{u}^T Q_{ij}\mathbf{u} + q_{ij}^T\mathbf{u} + g_{ij} + 2\rho \le 0 \tag{18}$$

From [7], we know a convex quadratic constraint can be converted into a second order cone constraint (see [7] for details). However, we can not gurantee the constraint in eq.(18) is a convex quadratic constraint since Q_{ij} in eq.(18) is not a positive semidefinite matrix.

Here we propose a new relaxation condition [3] under which the constraint in eq.(18) can be converted into a second order cone constraint. Suppose $\lambda_{ij}^1, \ldots, \lambda_{ij}^{D\mathcal{K}}$ be $D\mathcal{K}$ eigenvalues of matrix Q_{ij}. Let $v_{ij}^1, \ldots, v_{ij}^{D\mathcal{K}}$ be $D\mathcal{K}$ eigenvectors of Q_{ij} where v_{ij}^k is the eigenvector corresponding to λ_{ij}^k and satisfying $\|v_{ij}^k\| = 1$ and $(v_{ij}^k)^T v_{ij}^t = 0$ $(k \ne t)$. Then we have

$$Q_{ij} = \sum_{k=1}^{D\mathcal{K}} \lambda_{ij}^k v_{ij}^k (v_{ij}^k)^T \tag{19}$$

Suppose $\mathcal{M}$ contains index of all the eigenvalues of Q_{ij}, with $\mathcal{M} = \{1, 2, \ldots, D\mathcal{K}\}$. We can rewrite Q_{ij} in eq.(19) as:

$$Q_{ij} = Q_{ij}^+ + \sum_{k\in\mathcal{M}, \lambda_{ij}^k<0} \lambda_{ij}^k v_{ij}^k (v_{ij}^k)^T \tag{20}$$

with $Q_{ij}^+ = \sum_{k\in\mathcal{M}, \lambda_{ij}^k\ge 0} \lambda_{ij}^k v_{ij}^k (v_{ij}^k)^T$. Obviously, Q_{ij}^+ is a positive semidefinite matrix.

Substituting eq.(20) to eq.(18), we derive the following two constraints which are equivalent to eq.(18):

$$\mathbf{u}^T Q_{ij}^+\mathbf{u} + \sum_{k\in K, \lambda_{ij}^k<0} \lambda_{ij}^k z_k + q_{ij}^T\mathbf{u} + g_{ij} + 2\rho \le 0 \tag{21}$$

with

$$z_k = \mathbf{u}^T v_{ij}^k (v_{ij}^k)^T\mathbf{u} \qquad (\forall k \in \mathcal{M} \text{ and } \lambda_{ij}^k < 0) \tag{22}$$

Obviously the constraint in eq.(21) is a convex quadratic constraint. However the constraint in eq.(22) is a non-convex nonlinear constraint. Here we propose to relax this non-convex constraint into a convex constraint by allowing z_k to be a free variable but lower-bounded by a quadratic function as follows:

$$\mathbf{u}^T v_{ij}^k (v_{ij}^k)^T\mathbf{u} \le z_k \qquad (\forall k \in \mathcal{M} \text{ and } \lambda_{ij}^k < 0) \tag{23}$$

Considering the constraint in eq.(9), it is possible to make the above relaxation tighter by adding a linear upper-bound for z_k as (refer to [8] for details):

$$z_k \le 2\tilde{\mu}_k^{(0)}\tilde{\mu}_k + r^2 - (\tilde{\mu}_k^{(0)})^2 \qquad (\forall k \in \mathcal{M} \text{ and } \lambda_{ij}^k < 0). \tag{24}$$

After the relaxation, **Problem** 1 can be converted into the following convex optimization problem:

Problem 2

$$\min_{\mathbf{u},\rho} -\rho \tag{25}$$

subject to:

$$\mathbf{u}^T Q_{ij}^+\mathbf{u} + \sum_{k\in\mathcal{M}, \lambda_{ij}^k<0} \lambda_{ij}^k z_k + q_{ij}^T\mathbf{u} + g_{ij} + 2\rho \le 0 \tag{26}$$

$$\mathbf{u}^T v_{ij}^k (v_{ij}^k)^T\mathbf{u} \le z_k \le 2\tilde{\mu}_k^{(0)}\tilde{\mu}_k + r^2 - (\tilde{\mu}_k^{(0)})^2 \quad (\forall k \in \mathcal{M}\ \lambda_{ij}^k < 0) \tag{27}$$

$$\|\mathbf{u} - \mathbf{u}^{(0)}\| \le r \tag{28}$$

$$\rho \ge 0 \tag{29}$$

for all $X_i \in S$ and $j \in \Omega$ and $j \ne W_i$.

Since Q_{ij}^+ is a positive semi-definite matrix, it can be decomposed as $Q_{ij}^+ = L_{ij}^+(L_{ij}^+)^T$. As shown in [8], the above convex optimization **Problem 2** can be equivalently converted into the following standard SOCP format:

Problem 3

$$\min_{\mathbf{u},\rho} \ -\rho \tag{30}$$

subject to:

$$\left\| \begin{pmatrix} 1 + \sum_{k\in\mathcal{M}, \lambda_{ij}^k<0} \lambda_{ij}^k z_k + q_{ij}^T\mathbf{u} + g_{ij} + 2\rho \\ 2(L_{ij}^+)^T\mathbf{u} \end{pmatrix} \right\| \le 1 - \sum_{k\in\mathcal{M}, \lambda_{ij}^k<0} \lambda_{ij}^k z_k - q_{ij}^T\mathbf{u} - g_{ij} - 2\rho \tag{31}$$

$$\left\| \begin{pmatrix} 1 - z_k \\ 2(v_{ij}^k)^T\mathbf{u} \end{pmatrix} \right\| \le 1 + z_k \qquad (\forall k \in \mathcal{M}\ \lambda_{ij}^k < 0) \tag{32}$$

$$z_k \le 2\tilde{\mu}_k^{(0)}\tilde{\mu}_k + r^2 - (\tilde{\mu}_k^{(0)})^2 \qquad (\forall k \in \mathcal{M}\ \lambda_{ij}^k < 0) \tag{33}$$

$$\|\mathbf{u} - \mathbf{u}^{(0)}\| \le r \tag{34}$$

$$\rho \ge 0 \tag{35}$$

for all $X_i \in S$ and $j \in \Omega$ and $j \ne W_i$.

Problem 3 is a standard SOCP problem, which can be solved efficiently by many SOCP algorithms. In problem 3, the optimization is carried out w.r.t. $\mathbf{u}$, ρ and z_k while L_{ij}^{+}, q_{ij},g_{ij} and λ_{ij}^{k} are constants calculated from training data, and r is a pre-set parameter. However, due to the relaxation in eq.(27), this SOCP problem is just an approximation to the original LME problem.

5. EXPERIMENTAL RESULTS

The proposed SOCP-based optimization method for LME has been evaluated on the TIDIGITS database for connected digit string recognition in string level [2]. Only adult portion of the corpus is used in our experiments. 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-skip. Acoustic feature vectors consist of standard 39 dimensions (12 MFCC's 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 LME/SOCP, we use the best MCE models as the initial models and only HMM mean vectors are re-estimated. In each iteration of LME, a number of competing string-level models are computed for each utterance in training set based on its N-best decoding results ($N = 5$). Then we select support tokens according to eq.(2) and obtain the optimal Viterbi sequence for each support token according to the recognition result. Then, the relaxed SOCP optimization , i.e. **Problem 3**, is conducted with respect to $\mathbf{u}$, ρ and z_k. At last, CDHMM means are updated based on the optimization solution $\boldsymbol{u}^*$. In this work, **Problem3** is solved by an SOCP optimization tool, *MOSEK 4.0*[6] running under Matlab.

In our experiments, the LME/SOCP method (denoted as *SOCP*) has been compared with the LME using gradient descent method in [2], denoted as *GRAD* and the LME using the SDP method in [4], denoted as *SDP*. We also include the ML and MCE baseline systems in the table for reference. In Table 1, we gives performance comparison on the TIDIGITS test set using various training methods. From the results, we can also see that the LME/SOCP method outperforms the simple gradient descent method (except 32-mix) and the LME/SOCP is relatively easy to run while the gradient descent 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 still achieves the best overall performance, especially for large models. However, if we compare the efficiency between LME/SDP and LME/SOCP, the LME/SOCP runs substantially faster and consumes much less memory during optimization. The CPU times which is needed to optimize the problem per iteration are listed in Table 2 for comparison. It is clear that the LME/SOCP method runs about 20-200 times faster than the LME/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 [4] 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 LME/SOCP is significantly smaller than the LME/SDP.

Table 1. String error rates (in %) of various training methods on the TIDIGITS test data.

	ML	MCE	GRAD	SDP	**SOCP**
1-mix	12.61	6.72	3.77	2.75	**2.68**
2-mix	5.26	3.94	1.70	1.24	**1.23**
4-mix	3.48	2.23	1.24	0.89	**0.98**
8-mix	1.94	1.41	0.87	0.68	**0.80**
16-mix	1.72	1.11	0.82	0.63	**0.71**
32-mix	1.34	0.90	0.66	0.53	**0.66**

Table 2. Average optimization time (in seconds) per iteration of LME/SDP and LME/SOCP in the TIDIGITIS task.

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

6. SUMMARY

In this paper, we proposed a second order cone programming (SOCP) method for large margin estimation (LME) of CDHMMs in speech recognition. The new optimization method has been demonstrated to be effective in a connected digit string recognition task in the TIDIGITS database. Comparing with the previously proposed SDP method, the proposed LME/SOCP method runs much more efficiently in terms of optimization times and memory consumption. This opens up a door for applying the LME training to a state-of-the-art speech recognition system which normally involves very large HMM models. Currently, we are extending the LME/SOCP method to some larger scale speech tasks and the results will be reported in the future.

7. REFERENCES

[1] X. Li, H. Jiang and C. Liu, "Large Margin HMMs for Speech Recognition," *Proc. of IEEE ICASSP'2005*, Pennsylvania, Mar. 2005.

[2] H. Jiang, X. Li and C. Liu, "Large Margin Hidden Markov Models for Speech Recognition," *IEEE Trans. on Audio, Speech and Language Processing*, pp.1584-1595, Vol. 14, No. 5, September 2006.

[3] S. Kim and M. Kojima, "Second Order Cone Programming Relaxation of Nonconvex Quadratic Optimization Problems," *Optimization Methods and Software*, pp.201-224, Vol. 15, No. 3-4, 2001.

[4] X. Li and H. Jiang, "Solving Large Margin Estimation of HMMs via Semidefinite Programming," *Proc. of 2006 International Conference on Spoken Language Processing (ICSLP'2006)*, Pittsburgh, USA, April 2006.

[5] S. J. Benson, Y. Ye and X. Zhang, "Solving Large-Scale Sparse Semidefinite Programs for Combinatorial Optimization," *SIAM Journal on Optimization*, 10(2), 2000, pp. 443-461.

[6] Mosek ApS Inc., *The Mosek optimization tools (User's manual and reference)*, Ver. 4, 2006. (available from http://www.mosek.com)

[7] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of Second Order Cone Programming," *Linear Algebra Appl.*, pp. 193-228, No. 284, 1998.

[8] Y. Yin, "A Study on Convex Relaxations for Large Margin Estimation of HMMs," *M.S. thesis*, Department of Computer Science and Engineering, York University, Canada, 2007.