7

A General Approximation-Optimization Approach to Large Margin Estimation of HMMs

Hui Jiang and Xinwei Li
York University, Toronto
Canada

1. Introduction

The most successful modeling approach to automatic speech recognition (ASR) is to use a set of hidden Markov models (HMMs) as the acoustic models for subword or whole-word speech units and to use the statistical N-gram model as language model for words and/or word classes in sentences. All the model parameters, including HMMs and N-gram models, are estimated from a large amount of training data according to certain criterion. It has been shown that success of this kind of data-driven modeling approach highly depends on the goodness of estimated models. As for HMM-based acoustic models, the dominant estimation method is the Baum-Welch algorithm which is based on the maximum likelihood (ML) criterion. As an alternative to the ML estimation, discriminative training (DT) has also been extensively studied for HMMs in ASR. It has been demonstrated that various DT techniques, such as maximum mutual information (MMI), minimum classification error (MCE) and minimum phone error (MPE), can significantly improve speech recognition performance over the conventional maximum likelihood (ML) estimation.

More recently, we have proposed the large margin estimation (LME) of HMMs for speech recognition (Li et al., 2005; Liu et al., 2005a; Li & Jiang, 2005; Jiang et al., 2006), where Gaussian mixture HMMs are estimated based on the principle of maximizing the minimum margin. From the theoretical results in machine learning (Vapnik, 1998), a large margin classifier implies a good generalization power and generally yields much lower generalization errors in new test data, as shown in support vector machine and boosting method. As in Li et al., 2005 and Li & Jiang, 2005, estimation of large margin CDHMMs turns out to be a constrained minimax optimization problem. In the past few years, several optimization methods have been proposed to solve this problem, such as *iterative localized optimization* in Li et al., 2005, *constrained joint optimization method* in Li & Jiang, 2005 and Jiang et al., 2006, and *semi-definite programming (SDP) method* in Li & Jiang, 2006a and Li & Jiang 2006b. In this paper, we present a general Approximation-optiMization (AM) approach to solve the LME problem of Gaussian mixture HMMs in ASR. Similar to the EM algorithm, each iteration of the AM method consists of two distinct steps: namely **A**-step and **M**-step. In **A**-step, the original LME problem is approximated by a simple convex optimization problem in a close proximity of initial model parameters. In **M**-step, the approximate convex optimization problem is solved by using efficient convex optimization algorithms.

Source: Robust Speech Recognition and Understanding, Book edited by: Michael Grimm and Kristian Kroschel,
ISBN 987-3-90213-08-0, pp.460, I-Tech, Vienna, Austria, June 2007

This paper is structured as follows. In section 2, we present the large margin estimation formulation for HMMs in speech recognition. In section 3, we explain the proposed AM approach under a general framework. Next, as two examples, we consider to apply the AM method to solve the LME of HMMs for ASR. In section 4, we use the so-called **V-approx** for the case where competing hypotheses are given as N-Best lists. In section 5, we use **E-approx** for the case where competing hypotheses are given as word graphs. At last, some final remarks are discussed in section 6.

2. Large Margin Estimation (LME) of HMMs for ASR

In ASR, we consider a joint probability distribution between any speech utterance *X* and any word *W*, i.e. *p(X, W)*. Depending on the problem of interest, a word W may be any linguistic unit, e.g., a phoneme, a syllable, a word, a phrase, even a sentence. Given a speech utterance *X*, a speech recognizer will choose the word *W* as output based on the following plug-in MAP decision rule (Jiang et al., 1999):

$$\begin{aligned} \hat{W} &= \arg\max_{W} p(W \mid X) = \arg\max_{W} p(W) \cdot p(X \mid W) \\ &= \arg\max_{W} p(W) \cdot p(X \mid \lambda_W) = \arg\max_{W} \mathcal{F}(X \mid \lambda_W) \end{aligned} \tag{1}$$

where λ_W denotes the composite HMM representing word *W* and $\mathcal{F}(X|\lambda_W)$ is called discriminant function of λ_W given *X*, which is normally calculated in the logarithm domain as $\mathcal{F}(X|\lambda_W) = \ln[p(W) \cdot p(X|\lambda_W)] = \ln p(W) + \ln p(X|\lambda_W)$. In this work, we are only interested in estimating HMM λ_W and assume language model used to calculate *p(W)* is fixed.

For a speech utterance X_i , assuming its true word identity as W_i, following Weston & Watkins, 1999 and Crammer & Singer, 2001 and Altun et al., 2003, the multi-class separation margin for X_i is defined as:

$$d(X_i) = \mathcal{F}(X_i|\lambda_{W_i}) - \max_{j \in \Omega \; j \neq W_i} \mathcal{F}(X_i|\lambda_j) \tag{2}$$

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

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

Obviously, if $d(X_i) \leq 0$, X_i will be incorrectly recognized by the current HMM set, denoted as Λ, which includes all HMMs in the recognizer. On the other hand, if $d(Xi) > 0$, X_i will be correctly recognized by the model set Λ.

Given a set of training data $\mathcal{T}$ = {X_1 , X_2, ··· X_T}, we usually know the true word identities for all utterances in $\mathcal{T}$, denoted as $\mathcal{L}$ = {W_1, W_2, ··· W_T }- Thus, we can calculate the separation margin (or margin for short hereafter) for every utterance in *T* based on the definition in eq. (2) or (3). According to the statistical learning theory (Vapnik, 1998), the generalization error rate of a classifier in new test sets is theoretically bounded by a quantity related to its margin. A large margin classifier usually yields low error rate in new test sets and it shows more robust and better generalization capability. Motivated by the large margin principle, even for those utterances in the training set which all have positive margin, we may still want to maximize the minimum margin to build an HMM-based large

margin classifier. In this paper, we will study how to estimate HMMs for speech recognition based on the principle of maximizing minimum margin. First of all, from all utterances in $\mathcal{T}$, we need to identify a subset of utterances $\mathcal{S}$ as:

$$\mathcal{S} = \{X_i \mid X_i \in \mathcal{T} \text{ and } 0 \leq d(X_i) \leq \epsilon\} \tag{4}$$

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

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

The HMM models, Λ, estimated in this way, are called large margin HMMs.
Considering eq. (3), large margin estimation of HMMs can be formulated as the following *maximin* optimization problem:

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

Note it is fine to include all training data into the support token set with a large value for *e* in eq. (4). However, this may significantly increase the computational complexity in the following optimization process and most of those data with large margin are usually inactive in the optimization towards maximizing the minimum one, especially when a gradual optimization method is used, such as gradient descent and other local optimization methods.
As shown in Li et al., 2005 and Liu et al., 2005a and Jiang et al., 2006, the above *maximin* optimization may become unsolvable for Gaussian mixture HMMs because its margin as defined in eq. (3) may become unbounded with respect to model parameters. As one possible solution to solve this problem, some additional constraints must be imposed to ensure the margin is bounded during optimization as in Li & Jiang, 2005, Jiang et al., 2006. As suggested in Li & Jiang, 2006a and Liu et al., 2007, a KL-divergence based constraint can be introduced for HMM parameters to bound the margin. The KL-divergence (KLD) is calculated between an HMM λ $(\lambda \in \Lambda)$and its initial value as follows:

$$\mathcal{D}(\Lambda \,||\, \Lambda^{(n)}) = \sum_{\lambda \in \Lambda} \mathcal{D}(\lambda \,||\, \lambda^{(n)}) \leq r^2 \tag{7}$$

or

$$\mathcal{D}(\lambda \,||\, \lambda^{(n)}) \leq r^2 \qquad (\lambda \in \Lambda) \tag{8}$$

where $\lambda^{(n)}$ denotes the initial model parameters and r^2 is a constant to control a trust region for large margin optimization which is centered at the initial models. Since the KLD constraints given in eq. (7) defines a closed and compact set, it is trivial to prove that the margin in eq. (3) is a bounded function of HMM parameters λ so that the *maximin* optimization in eq. (6) is solvable under these constraints.

Furthermore, as shown in Li, 2005 and Li & Jiang, 2006a, the *maximin* optimization problem in eq. (6) can be equivalently converted into a constrained maximization problem by introducing a new variable ρ ($\rho > 0$) as a common lower bound to represent *min* part of all terms in eq.(6) along with the constraints that every item must be larger than or equal to ρ. As the result, the *maximin* optimization in eq.(6) can be equivalently transformed into the following optimization problem:

Problem 1

$$\tilde{\Lambda} = \arg\max_{\Lambda,\ \rho} \quad \rho \tag{9}$$

subject to:

$$\mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_j) \geq \rho \qquad \text{for all } X_i \in \mathcal{S} \text{ and } j \in \Omega \text{ and } j \neq W_i \tag{10}$$

$$\mathcal{D}(\Lambda \mid\mid \Lambda^{(n)}) = \sum_{\lambda \in \Lambda} \mathcal{D}(\lambda \mid\mid \lambda^{(n)}) \leq r^2 \tag{11}$$

$$\rho \geq 0 \tag{12}$$

3. A General Approximation-optiMization (AM) Approach to Large Margin Estimation of HMMs

Obviously, we can use a gradient descent method to solve the optimization Problem 1. As in Li & Jiang, 2005 and Jiang et al., 2006, we cast all constraints in eqs.(10) to (12) as some penalty terms in the objective function so that the model parameter updating formula can be easily derived. Thus, HMM parameters can be optimized gradually by following the calculated gradient direction. The gradient descent method is easy to implement and applicable to any differentiable objective functions. However, the gradient descent method can be easily trapped into a shallow local optimum if the derived objective function is jagged and complicated in nature, especially when the gradient descent method is operated in a high-dimensionality space. Also it is very difficult to set appropriate values for some critical parameters in gradient descent, such as step size and so on. As the result, the gradient method normally can not significantly improve over the initial models in a high-dimensionality space if the initial models have been set to some reasonably good values.

In this paper, we propose a novel approach to solve the above large margin estimation problem for HMMs. The key idea behind this approach is that we first attempt to find a simpler convex function to approximate the original objective function in a close proximity of initial model parameters if the original objective function is too complicated to optimize directly. Then, the approximate function is optimized by using an efficient convex optimization algorithm. In some cases, some relaxation must be made to ensure the resultant problem is indeed an convex optimization problem. As we know, a convex optimization problem can be efficiently solved even in a very high-dimensionality space

since it never suffers from the local optimum problem in a convex optimization problem. Based on the proximity approximation, we hope the optimal solution found for this approximate convex problem will also improve the original objective function as well. Then, in next iteration, the original objective function can be similarly approximated in the close proximity of this optimal solution as another convex optimization problem based on the same approximation principle. This process repeats until convergence conditions are met for the original objective function. Analogous to the popular EM algorithm (Dempster et al., 1977 and Neal & Hinton, 1998), each iteration consists of two separate steps: i) Approximation step (**A**-step): the original objective function is approximated in a close proximity of initial model parameters; ii) optiMization step (**M**-step): the approximate function is optimized by a convex optimization algorithm (convex relaxation may be necessary for some models). Analogously, we call this method as the AM algorithm. It is clear that the AM algorithm is more general than the EM algorithm since the expectation (**E**-step) can also be viewed as a proximity approximation method as we will show later. More importantly, comparing with the EM algorithm, the AM algorithm will be able to deal with more complicated objective functions such as those arising from discriminative training of many statistical models with hidden variables.

As one particular application of the AM algorithm, we will show how to solve the large margin estimation (LME) problem of HMMs in speech recognition.

3.1 Approximation Step (A-step):

There are many different methods to approximate an objective function in a close proximity. In this section, we introduce two different methods, namely Viterbi-based approximation **(V-approx)** and Expectation-based approximation **(E-approx).**

Let us first examine the log-likelihood function of HMMs, $\ln p(X|\lambda)$. Since HMMs have hidden variables such as unobserved state sequence, denoted as **s**, and unobserved Gaussian mixture labels (for Gaussian mixture HMMs), denoted as **1**, we have the following:

$$\ln p(X|\lambda) = \ln \sum_{\mathbf{s},\mathbf{l}} p(X, \mathbf{s}, \mathbf{l} \mid \lambda) \tag{13}$$

As the first way to approximate the above log-likelihood function, we can use the Viterbi approximation, i.e., we use the best Viterbi path, **s*** and **1***, to approximate the above summation instead of summing over all possible paths. And the best Viterbi path can be easily derived based on the initial models, $\lambda^{(n)}$ by the following *max* operation using the well-known Viterbi algorithm:

$$\{\mathbf{s}^*, \mathbf{l}^*\} = \arg\max_{\mathbf{s},\mathbf{l}} \; p(X, \mathbf{s}, \mathbf{l} \mid \lambda^{(n)}) \tag{14}$$

Thus, the log-likelihood function can be approximated as follows:

$$\ln p(X|\lambda) \approx \mathcal{V}(\lambda) = \ln p(X, \mathbf{s}^*, \mathbf{l}^*|\lambda) \tag{15}$$

This approximation scheme is named as Viterbi approximation, i.e., **V-approx.** Obviously, for HMMs, the approximate function $\mathcal{V}(\lambda)$ is a convex function.

If we assume the language model, *p(W),* is fixed, the discriminant function of HMM, $\mathcal{F}(\Lambda)$, in LME can be viewed as difference of log-likelihood functions. If we use the **V-approx** for both correct model $p(X_i|\lambda_{W_i})$ and incorrect competing model $p(X_i|\lambda_j)$:

$$\ln p(X_i|\lambda_{W_i}) \approx \mathcal{V}_i^{+}(\lambda_{W_i}) \tag{16}$$

$$\ln p(X_i|\lambda_j) \approx \mathcal{V}_i^{-}(\lambda_j) \tag{17}$$

Then, the constraints in eq.(10) can be approximated by difference of two convex functions as follows:

$$\mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_j) \approx \mathcal{V}_{ij}(\Lambda) = \mathcal{V}_i^{+}(\lambda_{W_i}) - \mathcal{V}_i^{-}(\lambda_j) \tag{18}$$

for all possible *i* and *j*.

Now let us consider the second scheme to approximate the objective function, i.e., **E-approx.** For the log-likelihood function of HMMs, $\ln p(X|\lambda)$ in eq.(13), we consider the following auxiliary function used in the EM algorithm:

$$\begin{aligned} \mathcal{Q}(\lambda|\lambda^{(n)}) &= \mathbb{E}_{\mathbf{s},\mathbf{l}}\left[\ln p(X,\mathbf{s},\mathbf{l} \mid \lambda) \;\middle|\; X,\lambda^{(n)}\right] \\ &= \sum_{\mathbf{s}}\sum_{\mathbf{l}} \ln p(X,\mathbf{s},\mathbf{l}|\lambda)\cdot \Pr(\mathbf{s},\mathbf{l}\,|X,\lambda^{(n)}) \end{aligned} \tag{19}$$

As shown in Dempster et al., 1977 and Neal & Hinton, 1998, the above auxiliary function is related to the original log-likelihood function as follows:

$$\mathcal{Q}(\lambda|\lambda^{(n)}) \le \ln p(X \mid \lambda) \tag{20}$$

$$\mathcal{Q}(\lambda|\lambda^{(n)})\Big|_{\lambda=\lambda^{(n)}} = \ln p(X|\lambda)\Big|_{\lambda=\lambda^{(n)}} \tag{21}$$

$$\frac{\partial\, \mathcal{Q}(\lambda|\lambda^{(n)})}{\partial\lambda}\Big|_{\lambda=\lambda^{(n)}} = \frac{\partial\, \ln p(X|\lambda)}{\partial\lambda}\Big|_{\lambda=\lambda^{(n)}} \tag{22}$$

From these, it is clear that $Q(\lambda|\lambda^{(n)})$ can be viewed as a close proximity approximation of log-likelihood function $\ln p(X|\lambda)$ at $\lambda^{(n)}$ with accuracy up to the first order. Under the proximity constraint in eq.(ll), it serves as a good approximation of the original log-likelihood function. Since the $\mathcal{Q}$ function is originally computed as an expectation in the EM algorithm, this approximation scheme is named as Expectation-based approximation **(E-approx).** Similarly, we use **E-approx** to approximate both correct model $p(X_i|\lambda_{W_i})$ and incorrect competing model $p(X_i|\lambda_j)$ in discriminant function as:

$$\ln p(X_i|\lambda_{W_i}) \approx \mathcal{Q}_i^{+}(\lambda_{W_i}|\lambda_{W_i}^{(n)}) \tag{23}$$

$$\ln p(X_i|\lambda_j) \approx \mathcal{Q}_i^{-}(\lambda_j|\lambda_j^{(n)}) \tag{24}$$

It can be easily shown that the $\mathcal{Q}$ function is also a convex function for HMMs. Therefore, the discriminant function can also be similarly approximated as difference of two convex functions as follows:

$$\mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_j) \approx \mathcal{Q}_{ij}(\Lambda) = \mathcal{Q}_i^{+}(\lambda_{W_i}|\lambda_{W_i}^{(n)}) - \mathcal{Q}_i^{-}(\lambda_j|\lambda_j^{(n)}) \tag{25}$$

for all possible *i* and *j*.

3.2 OptiMization Step (M-step):

After the approximation, either **V-approx** or **E-approx,** the original **Problem 1** has been converted into a relatively simpler optimization problem since all constraints have been approximately represented by differences of convex functions. As we know, a difference of two convex functions is not necessarily a convex function. Thus, in some cases, we will have to make some convex relaxations to convert the problem into a convex optimization problem so that a variety of convex optimization algorithms can be applied to find the global optimum of the approximate problem under the proximity constraint in eq.(ll). Due to this proximity constraint, we expect the global optimal solution will also improve the original LME optimization problem since the approximate convex optimization problem approaches the original LME problem with sufficient accuracy under the proximity constraint.

In the remainder of this paper, we will use two examples to show how to make convex relaxations and how to perform the convex optimization for LME of Gaussian mean vectors in Gaussian mixture HMMs. In the first example, we use **V-approx** to approximate the likelihood function of HMMs and then make some convex relaxations to convert the problem into an SDP (semi-definite programming) problem. This method is suitable for both isolated word recognition and continuous speech recognition based on the string-model of N-Best lists. In the second example, we use **E-approx** to approximate likelihood function of HMMs when the competing hypotheses are represented by word graphs or lattices. Then we similarly convert the problem into an SDP problem by making the same relaxation. This method is suitable for LME in large vocabulary continuous speech recognition where competing hypotheses are encoded in word graphs or lattices.

4. LME of Gaussian Mixture HMMs based on N-Best Lists

In this section, we apply the AM algorithm to solve the large margin estimation for Gaussian mixture HMMs in speech recognition. Here, we consider to use **V-approx** in the **A**-Step of the AM algorithm. This method is applicable to isolated word recognition and continuous speech recognition using string-models based on N-Best lists.

At first, we assume each speech unit, e.g., a word *W*, is modeled by an *N*-state Gaussian mixture HMM with parameter vector $\lambda = (\pi, A, \theta)$, where λ is the initial state distribution, $A = \{a_{ij} | 1 \le i, j \le N\}$ is transition matrix, and θ is parameter vector composed of mixture parameters $\theta_i = \{\omega_{ik}, \boldsymbol{\mu}_{ik}, \Sigma_{ik}\}_{k=1,2,\cdots,K}$ for each state *i*, where *K* denotes number of Gaussian mixtures in each state. The state observation p.d.f. is assumed to be a mixture of multivariate Gaussian distribution:

$$\begin{aligned} p(\mathbf{x}|\theta_i) &= \sum_{k=1}^{K} \omega_{ik} \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{ik}, \Sigma_{ik}) \\ &= \sum_{k=1}^{K} \omega_{ik} \cdot (2\pi)^{-D/2} |\Sigma_{ik}|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{ik})^T \Sigma_{ik}^{-1} (\mathbf{x}-\boldsymbol{\mu}_{ik})\right] \end{aligned} \quad (26)$$

where *D* denotes dimension of feature vector **x** and mixture weights ω_{ik}'s satisfy the constraint $\sum_{k=1}^{K} \omega_{ik} = 1$. In this paper, we only consider multivariate Gaussian

distribution with diagonal covariance matrix. Thus, the above state observation p.d.f. is simplified as:

$$p(\mathbf{x}|\theta_i) = \sum_{k=1}^{K} \omega_{ik}\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{ik}, \Sigma_{ik}) = \sum_{k=1}^{K} \omega_{ik} \prod_{d=1}^{D} \sqrt{\frac{1}{2\pi\sigma_{ikd}^2}} e^{-\frac{(x_d-\mu_{ikd})^2}{2\sigma_{ikd}^2}} \tag{27}$$

We assume training data is given as $\mathcal{T} = \{X_1, X_2, \cdots, X_T\}$ along with true labels for all utterances in $\mathcal{T}$, denoted as $\mathcal{L} = \{W_1, W_2, \cdots, W_T\}$. In this section, we assume for each *Xi* in $\mathcal{T}$, its competing hypotheses are encoded as an N-Best list, Ω_i, which can be generated from an N-Best Viterbi decoding process. We also assume the correct label has been excluded from the list. Then, the LME formulation in section 2 can be easily extended to this case. The only difference is that the set of all possible words, Ω, used to define margin for each training data X_i in eq.(2), becomes different for different training data, X_i, where we denote its N-Best list as Ω_i. And each model λ_{W_i} or λ_j denotes the string model concatenated according to the true transcription or a hypothesis from N-Best lists.

4.1 A-Step: V-approx

Given any speech utterance $X_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{iR}\}$, let $\mathbf{s} = \{s_1, s_2, \cdots, s_R\}$ be the unobserved state sequence, and $\mathbf{l} = \{l_1, l_2, \cdots, l_R\}$ be the associated sequence of the unobserved mixture component labels, if we use **V-approx,** the discriminant function, i.e., $\mathcal{F}(X_i|\lambda_j)$, can be expressed as:

$$\begin{aligned} \mathcal{F}(X_i|\lambda_j) \quad &\approx \quad \mathcal{V}_i(\lambda_j) = \log \pi_{s_1^*} + \sum_{t=2}^{R} \log a_{s_{t-1}^* s_t^*} + \sum_{t=1}^{R} \log \omega_{s_t^* l_t^*} \\ &- \quad \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}\Big[\log \sigma_{s_t^* l_t^* d}^2 + \frac{(x_{itd} - \mu_{s_t^* l_t^* d})^2}{\sigma_{s_t^* l_t^* d}^2}\Big] \end{aligned} \tag{28}$$

where we denote the optimal Viterbi path as $\mathbf{s}^* = \{s_1^*, s_2^*, \cdots, s_R^*\}$ and the best mixture component label as $\mathbf{l}^* = \{l_1^*, l_2^*, \cdots, l_R^*\}$.

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

$$\mathcal{V}_i(\lambda_j) = c_j^{"} - \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}\frac{(x_{itd} - \mu_{s_t^* l_t^* d})^2}{\sigma_{s_t^* l_t^* d}^2} \tag{29}$$

where c • is a constant which is independent from all Gaussian mean vectors.

Furthermore, we assume there are totally *M* Gaussian mixtures in the whole HMM set Λ, denoted as $\mathcal{M} = \{1, 2, \ldots, M\}$. We denote each Gaussian as $\mathcal{N}(u_k, \Sigma_k)$ where $k \in \mathcal{M}$. For notation convenience, the optimal Viterbi path **s*** and **1*** can be equivalently represented as a sequence of Gaussian mixture index, i.e., $\mathbf{j} = \{j_1, j_2, \cdots, j_R\}$, where $j_t \in \mathcal{M}$. is index

of Gaussians along the optimal Viterbi path {**s***, **1***}. Therefore, we can rewrite the discriminant function in eq. (29) according to this new Gaussian index as:

$$\mathcal{V}_i(\lambda_j) = c_j^{"} - \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}\frac{(x_{itd}-\mu_{j_t d})^2}{\sigma_{j_t d}^2} \tag{30}$$

For $\mathcal{F}(X_i|\lambda_{W_i})$, let us assume the optimal Viterbi path is $\mathbf{i} = \{i_1, i_2, \cdots, i_R\}$, where $i_t \in \mathcal{M}$. As we are only considering to estimate mean vectors of CDHMMs, after **V-approx,** the decision margin $d_{ij}(X_i)$ can be represented as a standard diagonal quadratic form as follows:

$$\begin{aligned} d_{ij}(X_i) &= \mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_j) \approx \mathcal{V}_i^{+}(\lambda_{W_i}) - \mathcal{V}_i^{-}(\lambda_j) \\ &= c_{ij} - \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}\Big[\frac{(x_{itd}-\mu_{i_t d})^2}{\sigma_{i_t d}^2} - \frac{(x_{itd}-\mu_{j_t d})^2}{\sigma_{j_t d}^2}\Big] \end{aligned} \tag{31}$$

where c_{ij} is another constant independent of all Gaussian means.
Furthermore, if we only estimate Gaussian mean vectors, the KL-divergence based constraint in eq.(11) can also be simplified for Gaussian mixture HMMs with diagonal covariance matrices as follows:

$$\mathcal{D}(\Lambda||\Lambda^{(n)}) = \sum_{k\in\mathcal{M}}\sum_{d=1}^{D}\frac{(\mu_{kd}-\mu_{kd}^{(n)})^2}{\sigma_{kd}^2} \le r^2 \tag{32}$$

To convert the above optimization problem into an SDP problem, we first represent the above approximated problem in a matrix form. We first define a mean matrix U by concatenating all normalized Gaussian mean vectors in Λ as its columns as follows:

$$U = (\tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\mu}}_2, \ldots, \tilde{\boldsymbol{\mu}}_M) \tag{33}$$

where each column is a normalized mean vector (column vector):

$$\tilde{\boldsymbol{\mu}}_k := (\frac{\mu_{k1}}{\sigma_{k1}};\ \frac{\mu_{k2}}{\sigma_{k2}};\ \ldots;\ \frac{\mu_{kD}}{\sigma_{kD}}) \tag{34}$$

Then we have

$$\begin{aligned} \mathcal{V}^{+}(\lambda_{W_i}) &= c_i' - \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}\frac{(x_{itd}-\mu_{i_t d})^2}{\sigma_{i_t d}^2} \\ &= c_i' - \frac{1}{2}\sum_{t=1}^{R}\sum_{d=1}^{D}(\tilde{x}_{itd}-\tilde{\mu}_{i_t d})^2 \\ &= c_i' - \frac{1}{2}\sum_{t=1}^{R}(\tilde{\boldsymbol{x}}_{i_t}-\tilde{\boldsymbol{\mu}}_{i_t})^T(\tilde{\boldsymbol{x}}_{i_t}-\tilde{\boldsymbol{\mu}}_{i_t}) \end{aligned} \tag{35}$$

where $\tilde{\boldsymbol{x}}_{i_t}$ denotes a normalized feature vector (column vector) as

$$\tilde{\boldsymbol{x}}_{i_t} := (\frac{x_{it1}}{\sigma_{i_t 1}};\ \frac{x_{it2}}{\sigma_{i_t 2}};\ \ldots;\ \frac{x_{itD}}{\sigma_{i_t D}}) \tag{36}$$

Since we have

$$\tilde{\boldsymbol{x}}_{i_t} - \tilde{\boldsymbol{\mu}}_{i_t} = (I_D, U)(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})$$

$$= \begin{pmatrix} \overbrace{\begin{matrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{matrix}}^{D} & \overbrace{\begin{matrix} \tilde{\mu}_{11} & \cdots & \tilde{\mu}_{L1} \\ \vdots & \vdots & \vdots \\ \tilde{\mu}_{1D} & \vdots & \tilde{\mu}_{LD} \end{matrix}}^{L} \end{pmatrix} \begin{pmatrix} \tilde{x}_{it1} \\ \vdots \\ \tilde{x}_{itD} \\ 0 \\ \vdots \\ -1 \quad (i_t) \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \} D \\ \\ \\ \} L \end{matrix} \tag{37}$$

where I_D is D-dimension identity matrix and $\boldsymbol{e}_k$ is a column vector with all zeros except only one -1 in k-th location. Then, we have

$$\begin{aligned} \mathcal{V}_i^+(\lambda_{W_i}) &= c_i' - \frac{1}{2}\sum_{t=1}^{R}(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})^T (I_D, U)^T (I_D, U)(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t}) \\ &= c_i' - \frac{1}{2}\sum_{t=1}^{R}(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})^T Z(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t}) \\ &= c_i' - \frac{1}{2}\sum_{t=1}^{R}\mathrm{tr}\left[(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})^T Z\right] \\ &= V_i \cdot Z + c_i' \end{aligned} \tag{38}$$

where V_i and Z are (D + L) x (D + L) dimensional symmetric matrices defined as:

$$V_i = -\frac{1}{2}\sum_{t=1}^{R}(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})(\tilde{\boldsymbol{x}}_{i_t}; \boldsymbol{e}_{i_t})^T \tag{39}$$

$$Z = (I_D, U)^T (I_D, U) = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \quad \text{with} \quad Y = U^T U \tag{40}$$

Similarly, we can express the discriminant function, $\mathcal{V}_i^-(\lambda_j)$, as:

$$\mathcal{V}_i^-(\lambda_j) = V_j \cdot Z + c_j''$$

where V_j is a (D + L) x (D + L) dimensional symmetric matrix defined as:

$$V_j = -\frac{1}{2}\sum_{t=1}^{R}(\tilde{\boldsymbol{x}}_{j_t}; \boldsymbol{e}_{j_t})(\tilde{\boldsymbol{x}}_{j_t}; \boldsymbol{e}_{j_t})^T \tag{41}$$

Thus, it is straightforward to convert the constraint in eq. (10) into the following form:

$$\mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\lambda_j) \approx \mathcal{V}_i^+(\lambda_{W_i}) - \mathcal{V}_i^-(\lambda_j) = V_{ij} \cdot Z - c_{ij} \geq \rho \tag{42}$$

where $V_{ij} = V_i - V_j$ and $c_{ij} = c_j'' - c_i'$.
Following the same line, we can convert the constraint in eq. (32) into the following matrix form as well:

$$\begin{aligned}
\mathcal{D}(\Lambda||\Lambda^{(n)}) &= \sum_{k\in\mathcal{M}}\sum_{d=1}^{D}(\tilde{\mu}_{kd} - \tilde{\mu}_{kd}^{(n)})^2 \\
&= \sum_{k\in\mathcal{M}}(\tilde{\boldsymbol{\mu}}_k - \tilde{\boldsymbol{\mu}}_k^{(n)})^T(\tilde{\boldsymbol{\mu}}_k - \tilde{\boldsymbol{\mu}}_k^{(n)}) \qquad (43)\\
&= \sum_{k\in\mathcal{M}}(\tilde{\boldsymbol{\mu}}_k^{(n)}; \boldsymbol{e}_k)^T(I_D; U)^T(I_D; U)(\tilde{\boldsymbol{\mu}}_k^{(n)}; \boldsymbol{e}_k) \qquad (44)\\
&= \sum_{k\in\mathcal{M}}\mathrm{tr}\left[(\tilde{\boldsymbol{\mu}}_k^{(n)}; \boldsymbol{e}_k)(\tilde{\boldsymbol{\mu}}_k^{(n)}; \boldsymbol{e}_k)^T\ Z\right] \\
&= R\cdot Z \le r^2 \qquad (45)
\end{aligned}$$

where R is a $(D + L)$ x $(D + L)$ dimensional symmetric matrix defined as:

$$R = \sum_{k\in\mathcal{M}}(\tilde{\boldsymbol{\mu}}_k^{(n)}; \boldsymbol{e}_k)(\tilde{\boldsymbol{\mu}}_k^{(n)}; \boldsymbol{e}_k)^T \qquad (46)$$

and $\tilde{\boldsymbol{\mu}}_k^{(n)}$ is normalized Gaussian mean vector in the initial model set, $\Lambda^{(n)}$ as defined in eq. (34).
In summary, after **V-approx** in the **A**-Step, the approximate optimization problem can be represented as:
Problem 2

$$\max_{Z,\,\rho}\quad \rho \qquad (47)$$

subject to:

$$V_{ij}\cdot Z - \rho \ge c_{ij} \qquad \text{for all } X_i \in \mathcal{S} \text{ and } j \in \Omega_i \qquad (48)$$

$$R\cdot Z \le r^2 \qquad (49)$$

$$Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \qquad \text{with} \qquad Y = U^TU \qquad (50)$$

$$\rho \ge 0 \qquad (51)$$

4.2 M-Step: SDP

Obviously, **Problem 2** is equivalent to the original LME optimization **Problem 1** except it is expressed in a matrix form. However, since the constraint $Y = U^TU$ is not convex, it is a non-convex optimization problem. Thus, some relaxations are necessary to convert it into a convex optimization problem. In this section, we consider to use a standard SDP (semi-definite programming) relaxation to convert **Problem 2** into an SDP problem.
As shown in Boyd et al., 1994, the following statement always holds for matrices:

$$Y - U^TU \succeq 0 \qquad \Leftrightarrow \qquad Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \succeq 0 \qquad (52)$$

where $Z \succeq 0$ denotes Z is a positive semidefinite matrix.
Therefore, following Boyd et al., 1994, if we relax the constraint $Y = U^T U$ to $Y - U^T U \succeq 0$, we are able to make Z a positive semidefinite matrix. During the optimization, the top left corner of Z must be an identity matrix, i.e., $Z_{1:D,1:D} = I_D$, which can be easily represented as a group of linear constraints as:

$$\left[(\boldsymbol{e}_k + \boldsymbol{e}_l)(\boldsymbol{e}_k + \boldsymbol{e}_l)^T\right] \cdot Z = 2 + 2 \cdot \delta(k - l) \qquad \text{for} \qquad 1 \le k \le D, \;\; k \le l \le D \tag{53}$$

where $\delta(k - l)$ is 1 when $k = l$ and 0 otherwise. If $k = l$ (for $l \le k$, $l \le D$),

$$\left[(\boldsymbol{e}_k + \boldsymbol{e}_l)(\boldsymbol{e}_k + \boldsymbol{e}_l)^T\right] \cdot Z = 4 z_{kk} = 4 \tag{54}$$

otherwise

$$\left[(\boldsymbol{e}_k + \boldsymbol{e}_l)(\boldsymbol{e}_k + \boldsymbol{e}_l)^T\right] \cdot Z = z_{kk} + z_{ll} + z_{kl} + z_{lk} = z_{kk} + z_{ll} + 2 z_{kl} = 2 \tag{55}$$

since Z is a symmetric matrix, $z_{kl} = z_{lk}$. Obviously, the unique solution for this set of linear constraints is $z_{kk} = 1$ and $z_{kl} = z_{lk} = 0$ for all $1 \le k \le D$, $k \le l \le D$.
Finally, under the relaxation in eq. (52), **Problem 2** is converted into a standard SDP problem as follows:
Problem 3

$$\max_{Z,\, \rho} \;\; \rho \tag{56}$$

subject to:

$$V_{ij} \cdot Z - \rho \ge c_{ij} \qquad \text{for all } X_i \in \mathcal{S} \text{ and } j \in \Omega_i \tag{57}$$

$$R \cdot Z \le r^2 \tag{58}$$

$$Z_{1:D,1:D} = I_D \tag{59}$$

$$Z \succeq 0 \qquad \text{and} \qquad \rho \ge 0 \tag{60}$$

Problem 3 is a standard SDP problem, which can be solved efficiently by many SDP algorithms, such as interior-point methods (Boyd & Vandenberghe, 2004). In **Problem 3,** the optimization is carried out w.r.t. Z (which is constructed from all HMM Gaussian means) and ρ, and V_{ij} and c_{ij} and R are constant matrix calculated from training data and initial models, and r is a pre-set control parameter. Then, all Gaussian mean vectors are updated based on the found SDP solution Z^*.
At last, the AM algorithm for LME of Gaussian mixture HMMs based on N-Best lists is summarized as follows:

Algorithm 1 The AM Algorithm for LME of HMMs based on N-Best Lists

repeat

1. Perform N-Best Viterbi decoding for all training data using models $\lambda^{(n)}$
2. Identify the support set $\mathcal{S}$ according to eq. (4).
3. **A-Step:** collect sufficient statistics including R, and V_{ij}, c_{ij} for all $X_i \in \mathcal{S}$ and $j \in \Omega_i$
4. **M-Step:** Perform SDP to solve **Problem 3** and update models.
5. $n = n + 1$.

until some convergence conditions are met.

5. LME of Gaussian mixture HMMs based on Word Graphs

As in most large vocabulary continuous speech recognition systems, competing hypotheses for training utterances are encoded in a more compact format, i.e., word graphs or word lattices. In this section, we apply the AM algorithm to LME of Gaussian mixture HMMs for speech recognition based on word graphs instead of N-Best lists. Here, we use **E-approx** in **A**-step to derive an efficient method to conduct LME for large vocabulary continuous speech recognition tasks using word graphs.

Assume we are given a training set as $\mathcal{T} = \{X_1, X_2, \cdots, X_T\}$, for each training utterance X_i, assume its true transcription is W_i and its competing hypotheses are represented as a word graph, denoted as $\mathcal{G}_i$. Ideally the true transcription Wj should be excluded from the word graph $\mathcal{G}_i$. In this case, we define the margin for X_i as follows:

$$\begin{aligned} d(X_i) &= \mathcal{F}(X_i|\lambda_{W_i}) - \mathcal{F}(X_i|\mathcal{G}_i) \\ &= \left[\ln p(X_i|\lambda_{W_i}) + \ln p(W_i)\right] - \ln \sum_{j \in \mathcal{G}_i} \left[p(X_i|\lambda_j) \cdot p(W_j)\right] \end{aligned} \tag{61}$$

where the summation is taken for all hypotheses in word graph $\mathcal{G}_i$. In this paper, we only consider to estimate acoustic models and assume language model scores $p(W_i)$ and $p(W_j)$ are constants. Then, the idea of LME in section 2 can be extended to estimate acoustic models towards maximizing the minimum margin across a selected support token set $\mathcal{S}$, as in eq.(5). In the following, we consider to solve this LME problem with the AM algorithm where **E-approx** is used in **A**-step and SDP is used to solve **M**-step. For simplicity, we only estimate Gaussian mean vectors and assume other HMM parameters are kept constant in LME. But it is quite trivial to extend to estimating all HMM parameters with the same idea.

5.1 A-Step: E-approx

Given any speech utterance $X_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \cdots, \mathbf{x}_{iR}\}$, in **E-approx,** the HMM log-likelihood function $\ln p(X_i|\lambda_{W_i})$ is approximated by the following auxiliary function calculated based on expectation:

$$\begin{aligned} \mathcal{Q}_i^+(\Lambda|\Lambda^{(n)}) &= \sum_{\mathbf{s}} \sum_{\mathbf{l}} \ln p(X_i, \mathbf{s}, \mathbf{l} \mid \lambda_{W_i}) \cdot \Pr(\mathbf{s}, \mathbf{l} \mid X_i, \lambda_{W_i}^{(n)}) \\ &= -\frac{1}{2} \sum_{k \in \mathcal{M}} \sum_{t=1}^{R} \sum_{d=1}^{D} \frac{(x_{itd} - \mu_{kd})^2}{\sigma_{kd}^2} \cdot \gamma_i(k,t) + b_i^* \end{aligned} \tag{62}$$

where $\gamma_i(k,t)$ denotes posterior probability calculated for k-th Gaussian component in the model set ($k \in \mathcal{M}$) using the Baum- Welch algorithm conditional on the initial model $\Lambda^{(n)}$ and training utterance X_i, and b_i^* is a constant independent from all Gaussian mean vectors.

After rearranging all terms in eq. (62), we can organize $\mathcal{Q}_i^+(\Lambda_i|\Lambda_i^{(n)})$ as a quadratic function of all Gaussian mean vectors:

$$\mathcal{Q}_i^+(\Lambda|\Lambda^{(n)}) = -\frac{1}{2} \sum_{k \in \mathcal{M}} \sum_{d=1}^{D} \xi_{ik} \cdot \left(\frac{\bar{\mathrm{x}}_{ikd} - \mu_{kd}}{\sigma_{kd}}\right)^2 + b_i' \tag{63}$$

where b_i' is another constant independent of Gaussian mean vectors and

$$\xi_{ik} = \sum_{t=1}^{R} \gamma_i(k,t) \tag{64}$$

$$\bar{\mathrm{x}}_{ikd} = \frac{\sum_{t=1}^{R} \gamma_i(k,t) \cdot x_{itd}}{\sum_{t=1}^{R} \gamma_i(k,t)} \tag{65}$$

If we denote two column vectors as

$$\tilde{\boldsymbol{\mu}}_k := \left(\frac{\mu_{k1}}{\sigma_{k1}}; \frac{\mu_{k2}}{\sigma_{k2}}; \cdots; \frac{\mu_{kD}}{\sigma_{kD}} \right) \qquad (\text{for } \; k \in \mathcal{M}) \tag{66}$$

and

$$\tilde{\mathbf{x}}_{ik} := \left(\frac{\bar{\mathrm{x}}_{ik1}}{\sigma_{k1}}; \; \frac{\bar{\mathrm{x}}_{ik2}}{\sigma_{k2}}; \; \cdots; \; \frac{\bar{\mathrm{x}}_{ikD}}{\sigma_{kD}} \right) \tag{67}$$

then we have

$$\begin{aligned} \mathcal{Q}_i^+(\Lambda|\Lambda^{(n)}) &= b_i' - \frac{1}{2} \sum_{k \in \mathcal{M}} \sum_{d=1}^{D} \xi_{ik} \cdot \left(\frac{\bar{\mathrm{x}}_{ikd} - \mu_{kd}}{\sigma_{kd}} \right)^2 \\ &= b_i' - \frac{1}{2} \sum_{k \in \mathcal{M}} \xi_{ik} \cdot (\tilde{\mathbf{x}}_{ik} - \tilde{\boldsymbol{\mu}}_k)^T (\tilde{\mathbf{x}}_{ik} - \tilde{\boldsymbol{\mu}}_k) \end{aligned} \tag{68}$$

Since we have

$$\begin{aligned} \tilde{\mathbf{x}}_{ik} - \tilde{\boldsymbol{\mu}}_k &= (I_D, U)(\tilde{\mathbf{x}}_{ik};\; \boldsymbol{e}_k) \\ &= \begin{pmatrix} \overbrace{\begin{matrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{matrix}}^{D} & \overbrace{\begin{matrix} \tilde{\mu}_{11} & \cdots & \tilde{\mu}_{L1} \\ \vdots & \vdots & \vdots \\ \tilde{\mu}_{1D} & \vdots & \tilde{\mu}_{LD} \end{matrix}}^{L} \end{pmatrix} \begin{pmatrix} \tilde{x}_{ik1} \\ \vdots \\ \tilde{x}_{ikD} \\ 0 \\ \vdots \\ -1 \;\; (k) \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} D \\ \\ \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\} L \end{matrix} \end{aligned} \tag{69}$$

Then, we have

$$\begin{aligned} \mathcal{Q}_i^+(\Lambda|\Lambda^{(n)}) &= b_i' - \frac{1}{2} \sum_{k \in \mathcal{M}} \xi_{ik} \cdot (\tilde{\mathbf{x}}_{ik}; \boldsymbol{e}_k)^T (I_D, U)^T (I_D, U)(\tilde{\mathbf{x}}_{ik}; \boldsymbol{e}_k) \\ &= b_i' - \frac{1}{2} \sum_{k \in \mathcal{M}} \xi_{ik} \cdot \mathrm{tr}\left[(\tilde{\mathbf{x}}_{ik}; \boldsymbol{e}_k)(\tilde{\mathbf{x}}_{ik}; \boldsymbol{e}_k)^T Z \right] \\ &= Q_i^+ \cdot Z + b_i' \end{aligned} \tag{70}$$

where Z and Q_i^+ are (D + L) x (D + L) dimensional symmetric matrices, Z is defined as in eq.(40) and Q_i^+ is calculated as:

$$Q_i^+ = -\frac{1}{2}\sum_{k\in\mathcal{M}} \xi_{ik}\cdot(\tilde{\mathbf{x}}_{ik};\boldsymbol{e}_k)(\tilde{\mathbf{x}}_{ik};\boldsymbol{e}_k)^T \tag{71}$$

Similarly, we consider to approximate the log-likelihood function of word graph with **E-approx** as follows:

$$\begin{aligned}
\mathcal{F}(X_i|\mathcal{G}_i) &= \ln\sum_{j\in\mathcal{G}_i}\Big[p(X_i|\lambda_j)\cdot p(W_j)\Big] \approx Q_i^-(\Lambda|\Lambda^{(n)}) \\
&= \sum_{j\in\mathcal{G}_i}\sum_{\mathbf{s}}\sum_{\mathbf{l}}\Big[\ln p(X_i,\mathbf{s},\mathbf{l}\mid\lambda_j)+\ln p(W_j)\Big]\cdot\Pr(\mathbf{s},\mathbf{l}\mid X_i,\Lambda^{(n)}) \\
&= -\frac{1}{2}\sum_{j\in\mathcal{G}_i}\sum_{k\in\mathcal{M}}\sum_{t=1}^{R}\sum_{d=1}^{D}\frac{(x_{itd}-\mu_{kd})^2}{\sigma_{kd}^2}\cdot\gamma_j(k,t)+b_i^* \\
&= -\frac{1}{2}\sum_{k\in\mathcal{M}}\sum_{d=1}^{D}\xi'_{ik}\cdot\left(\frac{\bar{\mathrm{x}}'_{ikd}-\mu_{kd}}{\sigma_{kd}}\right)^2+b_i^{"}
\end{aligned} \tag{72}$$

where both b_i^* and $b_i^{"}$ are two constants independent of Gaussian mean vectors, and

$$\xi'_{ik} = \sum_{j\in\mathcal{G}_i}\sum_{t=1}^{R}\gamma_j(k,t) \tag{73}$$

$$\bar{\mathrm{x}}'_{ikd} = \frac{\sum_{j\in\mathcal{G}_i}\sum_{t=1}^{R}\gamma_j(k,t)\cdot x_{itd}}{\sum_{j\in\mathcal{G}_i}\sum_{t=1}^{R}\gamma_j(k,t)} \tag{74}$$

And ξ'_{ik} and $\bar{\mathrm{x}}'_{ikd}$ can be calculated efficiently by running the forward-backward algorithm in the word graph $\mathcal{G}_i$ as in Wessel et al., 2001.
Similarly, $\mathcal{F}(X_i|\mathcal{G}_i)$ can be expressed as the following matrix form:

$$\mathcal{F}(X_i|\mathcal{G}_i) = Q_i^-\cdot Z + b_i^{"} \tag{75}$$

where

$$Q_i^- = -\frac{1}{2}\sum_{k\in\mathcal{M}}\xi'_{ik}\cdot(\tilde{\mathbf{x}}'_{ik};\ \boldsymbol{e}_k)(\tilde{\mathbf{x}}'_{ik};\ \boldsymbol{e}_k)^T \tag{76}$$

Therefore, the margin *d(Xi)* can be approximated as:

$$d(X_i) \approx \mathcal{Q}_i^+(\Lambda|\Lambda^{(n)}) - \mathcal{Q}_i^-(\Lambda|\Lambda^{(n)}) = Q_i^{\mathrm{d}}\cdot Z - b_i \tag{77}$$

where $Q_i^{\mathrm{d}} = Q_i^+ - Q_i^-$ and $b_i = b_i^{"} - b'_i$.
As the result, the LME problem based on the word graphs can be approximately represented as follows:

Problem 4

$$\max_{Z,\,\rho} \quad \rho \tag{78}$$

subject to::

$$Q_i^{\mathrm{d}} \cdot Z - \rho \geq b_i \qquad \text{for all} \quad X_i \in \mathcal{S} \tag{79}$$

$$R \cdot Z \leq r^2 \tag{80}$$

$$Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \qquad \text{with} \qquad Y = U^T U \tag{81}$$

$$\rho \geq 0 \tag{82}$$

5.2 M-Step: SDP

Next, using the same relaxation in eq.(52), we convert the above optimization problem into the following SDP problem:

Problem 5

$$\max_{Z,\,\rho} \quad \rho \tag{83}$$

subject to::

$$Q_i^{\mathrm{d}} \cdot Z - \rho \geq b_i \qquad \text{for all} \quad X_i \in \mathcal{S} \tag{84}$$

$$R \cdot Z \leq r^2 \tag{85}$$

$$Z_{1:D,1:D} = I_D \tag{86}$$

$$Z \succeq 0 \qquad\qquad \rho \geq 0 \tag{87}$$

The **Problem 5** can be efficiently solved using a fast SDP optimization algorithm. The found solution Z^* can be used to update all Gaussian mean vectors.

At last, the AM algorithm for LME of Gaussian mixture HMMs using **E-approx** based on word graphs is summarized as follows:

Algorithm 2 The AM Algorithm for LME of HMMs based on Word Graphs

repeat

1. Perform Viterbi decoding for all training data to generate word graphs using models $\Lambda^{(n)}$.
2. Identify the support set $\mathcal{S}$ according to eq.(4).
3. **A-Step:** collect sufficient statistics including R, and Q_i^{d}, b_i for all $X_i \in \mathcal{S}$.
4. **M-Step:** Perform SDP to solve **Problem 5** and update models.
5. $n = n + 1$.

until some convergence conditions are met.

6. Final Remarks

In this paper, we have proposed a general Approximation-optiMization (AM) approach for large margin estimation (LME) of Gaussian mixture HMMs in speech recognition. Each iteration of the AM method consists of A-step and M-step. In A-step, the original LME problem is approximated by a simple convex optimization problem in a close proximity of initial model parameters. In M-step, the approximate convex optimization problem is solved by using efficient convex optimization algorithms. The AM method is a general approach which can be easily applied for discriminative training of statistical models with hidden variables. In this paper, we introduce two examples to apply the AM approach to LME of Gaussian mixture HMMs. The first method uses V-approx and is applicable for isolated word recognition and continuous speech recognition based on N-Best lists. The second method uses E-approx and can be applied to large vocabulary continuous speech recognition when competing hypotheses are given as word graphs or word lattices. Due to space limit, we can not report experimental results in this paper. Readers can refer to Li, 2005 and Li & Jiang, 2006a, Li & Jiang, 2006b for details about ASR experiments.

7. References

Altun, Y.; Tsochantaridis, I. & Hofmann, T. (2003). Hidden Markov Support Vector Machines, *Proc. of the 20th International Conference on Machine Learning (ICML-2003),* Washington B.C..

Arenas-Garcia, J. & Perez-Cruz, F. (2003). Multi-class support vector machines: a new approach, *Proc. of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2003),* pp.II-781-784.

Boyd, S.; Ghaoui, L. E.; Feron, E. & Balakrishnan, V. (1994). Linear matrix inequalities in system and control theory, *Society for Industrial and Applied Mathematics (SIAM),* Philadelphia, PA.

Boyd, S & Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

Crammer, K. & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines, *Journal of Machine Learning Research,* Vol. 2, pp.265-292.

Dempster, A. P.; Laird, N. M. & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statistical Society B,* Vol. 39, pp. 1-38.

Jiang, EL; Hirose, K. & Huo, Q. (1999). Robust speech recognition based on Bayesian prediction approach, *IEEE Trans, on Speech and Audio Processing,* pp. 426-440, Vol. 7, No.4.

Jiang, H. (2004). Discriminative Training for Large Margin HMMs, *Technical Report CS-2004-01,* Department of Computer Science and Engineering, York University.

Jiang, H.; Soong, F. & Lee, C.-H. (2005). A dynamic in-search data selection method with its applications to acoustic modeling and utterance verification, *IEEE Trans, on Speech and Audio Processing,* pp.945-955, Vol. 13, No.5.

Jiang, H.; Li, X. & Liu, C.-J. (2006). Large Margin Hidden Markov Models for Speech Recognition, *IEEE Trans, on Audio, Speech and Language Processing,* pp.1584-1595, Vol. 14, No. 5.

Jiang, H. & Li, X. (2007). Incorporating Training Errors for Large Margin HMMs under Semi-definite Programming Framework, *Proc. of 2007 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2007),* Hawaii, USA.

Li, X.; Jiang, H. & Liu, C.-J. (2005). Large margin HMMs for speech recognition, *Proc. of 2005 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2005),* pp.V513-516, Philadelphia, Pennsylvania.

Li, X. & Jiang, H. (2005). A constrained joint optimization method for large margin HMM estimation, *Proc. of 2005 IEEE workshop on Automatic Speech Recognition and Understanding.*

Li, X. (2005). Large Margin Hidden Markov Models for Speech Recognition. *M.S. thesis,* Department of Computer Science and Engineering, York University, Canada.

Li, X. & Jiang, H. (2006a). Solving Large Margin HMM Estimation via Semi-definite Programming, *Proc. of 2006 International Conference on Spoken Language Processing (ICSLP'2006),* Pittsburgh, USA.

Li, X. & Jiang, H. (2006b). Solving Large Margin Hidden Markov Model Estimation via Semidefinite Programming, *submitted to IEEE Trans, on Audio, Speech and Language Processing.*

Liu, C.-J.; Jiang, H. & Li, X. (2005a). Discriminative training of CDHMMs for Maximum relative separation margin, *Proc. of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2005),* pp.V101-104, Philadelphia, Pennsylvania.

Liu, C.-J.; Jiang, H. & Rigazio, L. (2005b) "Maximum relative margin estimation of HMMs based on N-best string models for continuous speech recognition," *Proc. of IEEE workshop on Automatic Speech Recognition and Understanding.*

Liu, C.; Liu, P.; Jiang, H.; Soong, F. & Wang, R.-H. (2007). A Constrained Line Search Optimization For Discriminative Training in Speech Recognition, *Proc. of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'2007),* Hawaii, USA.

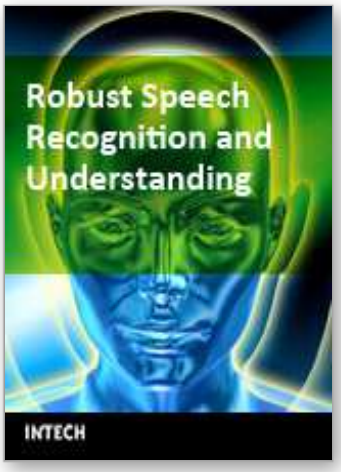
Neal, R. & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants, *in M. I. Jordan (Ed.), Learning in Graphical Models,* pp.355-368, Kluwer Academic Publishers.

Smola, A. J.; Bartlett, P.; Scholkopf, B. & Schuurmans, D. (ed.) (2000). *Advances in Large Margin Classifiers,* the MIT Press.

Wessel, F.; Schluter, R.; Macherey, K. & Ney, H. (2001). Confidence measures for large vocabulary continuous speech recognition. *IEEE Trans, on Speech and Audio Processing,* Vol. 9, No. 3, 288-298.

Weston, J. & Watkins, C. (1999). Support vector machines for multi-class pattern recognition, *Proc. of European Symposium on Artificial Neural Networks.*

Vapnik, V. N. (1998) *Statistical Learning Theory,* Wiley, 1998.



Robust Speech Recognition and Understanding

Edited by Michael Grimm and Kristian Kroschel

ISBN 978-3-902613-08-0
Hard cover, 460 pages
Publisher I-Tech Education and Publishing
Published online 01, June, 2007
Published in print edition June, 2007

This book on Robust Speech Recognition and Understanding brings together many different aspects of the current research on automatic speech recognition and language understanding. The first four chapters address the task of voice activity detection which is considered an important issue for all speech recognition systems. The next chapters give several extensions to state-of-the-art HMM methods. Furthermore, a number of chapters particularly address the task of robust ASR under noisy conditions. Two chapters on the automatic recognition of a speaker's emotional state highlight the importance of natural speech understanding and interpretation in voice-driven systems. The last chapters of the book address the application of conversational systems on robots, as well as the autonomous acquisition of vocalization skills.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821