INCORPORATING TRAINING ERRORS FOR LARGE MARGIN HMMS UNDER SEMI-DEFINITE PROGRAMMING FRAMEWORK

Hui Jiang, Xinwei Li

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

ABSTRACT

In this paper, we study how to incorporate training errors in large margin estimation (LME) under semi-definite programming (SDP) framework. Like soft-margin SVM, we propose to optimize a new objective function which linearly combines the minimum margin among positive tokens and an average error function of all negative tokens. The new method is named as *soft-LME*. It is shown the new *soft-LME* problem can still be converted into an SDP problem if we properly define the average error function of all negative tokens based on their discriminative functions. Some preliminary results on TIDIGITS show that the soft-LME/SDP method yields modest performance gain when training error rates are significant. Moreover, it is also shown that the soft-LME/SDP can achieve much faster convergence for all cases which we have investigated.

Index Terms: large margin estimation (LME), soft-LME, CDHMM, semi-definite programming (SDP), discriminative training.

1. INTRODUCTION

Recently, we have proposed the large margin estimation (LME) method for speech recognition [5, 4], where Gaussian mixture continuous density hidden Markov models (CDHMM) are estimated based on the large margin principle. As shown in [5, 4], estimation of large margin CDHMMs turns out to be a minimax optimization problem. Many optimization methods have been used for LME, ranging from simple gradient decent method in [5, 4] to more advanced semi-definite programming (SDP) approach in [7, 6]. In [7, 6], we have formulated the LME problem of Gaussian mixture CDHMMs into an SDP problem under some relaxation conditions so that the LME can be solved with a variety of efficient optimization algorithms which are particularly designed for SDP, such as interior-point methods. As in [7], the LME/SDP method has been successfully applied to some small to medium size speech recognition tasks using ISOLET and TIDIGITS databases. Our previous study clearly shows that the LME/SDP framework is superior to other optimization methods for LME in both theoretical elegance and practical effectiveness. But, as in most large margin classifiers, a meaningful margin can only be defined for those correctly classified tokens (the so-called positive tokens) in training set. Because of this, in the original LME formulation [4], we propose to use only a subset of positive tokens (the so-called support tokens) in LME training and we simply ignore all mis-recognized data (the so-called negative tokens) in each step of LME training. Although we have observed that many negative tokens may become positive after each iteration of LME so that they will be eventually used for LME in next iteration. However, this may become a very serious problem when we apply the LME method to large-vocabulary continuous speech recognition (LVCSR) tasks since the training error rates remains relatively high in many LVCSR tasks, especially when measured in string level. This problem has been first addressed in [8] for a variant of large margin criterion, namely maximum relative margin estimation (MRME), using a simple gradient descent optimization method. In [8], two methods have been used to solve this problem: i) optimize a new objective function which combines both minimum relative margin and training errors; ii) directly include negative tokens in MRME. Both methods are shown to be effective in terms of improving recognition performance.

In this paper, we will study how to incorporate training errors for LME under the SDP framework. Following the same idea of extending support vector machine (SVM) from linearly separable cases to non-separable cases, we consider to optimize a new objective function which linearly combines the minimum margin among positive tokens and an average error function of all negative tokens. This new estimation criterion is named as *soft Large Margin Estimation* (*soft-LME*) in this paper. As we will show later, the new *soft-LME* problem can still be converted into an SDP problem as long as we properly define the average error function of all negative tokens based on their discriminative functions. In this way, all efficient SDP algorithms can still be used to solve the *soft-LME* problem. In this work, the proposed soft-LME/SDP framework has been investigated in a connected digit string recognition task using the TIDIGITS database. Experimental results show the soft-LME/SDP yields some modest performance gain for those models where training error rates are significant. Moreover, it is also shown that the soft-LME/SDP can achieve much faster convergence for all cases which we have investigated.

2. LARGE MARGIN ESTIMATION OF CDHMM

From [5, 4], we know that the separation margin for a speech utterance X_i in a multi-class classifier can be defined as:

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

where Ω denotes the set of all possible words, λ_W denotes the HMM representing the word W, W_i is the true word identity for X_i and $\mathcal{F}(X|\lambda_W)$ is called discriminant function. Usually, the discriminant function is calculated in the logarithm scale: $\mathcal{F}(X|\lambda_W) = \log\left[p(W)\cdot p(X|\lambda_W)\right]$. 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 threshold. 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.

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

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

As shown in [5, 4], the margin as defined in eq.(1) is actually unbounded for Gaussian mixture CDHMMs. Following [6, 7], we introduce a constraint based on KL divergence between model parameters and their initial values:

$$R(\Lambda) = \sum_i \mathcal{D}(\lambda_i \mid\mid \lambda_i^{(0)}) \leq r^2 \tag{4}$$

where $\mathcal{D}$ denotes KL-divergence. Therefore, large margin estimation (LME) of CDHMMs essentially is a constrained minimax optimization problem. As shown in [6, 7], it can eventually be converted into a semi-definite programing problem under some relaxation conditions.

3. INCORPORATING TRAINING ERRORS IN LME

Obviously, in the above LME formulation, we only use the support token set, $\mathcal{S}$, which only includes correctly recognized tokens (i.e., positive tokens) in training set. And all mis-recognized training data (i.e., negative tokens) are simply discarded in each iteration of LME. This may affect effectiveness and efficiency of LME training in large vocabulary ASR tasks, where we typically have quite high training error rates. In this work, following the same idea of extending support vector machine (SVM) from linearly separable cases to non-separable cases, we propose to extend the original LME/SDP method in [7] to incorporate training errors as well. As we will show later, the resultant problem can still be converted into a semi-definite programming problem so that many efficient optimization methods are still applicable.

First of all, let's define an error set, $\mathcal{E}$, which contains all negative tokens in training data:

$$\mathcal{E} = \{X_i \mid X_i \in \mathcal{D} \text{ and } d(X_i) < 0\} \tag{5}$$

Like the soft-margin SVM for non-separable cases in [2], a new large margin estimation criterion to incorporate training errors can be defined as a linear combination of minimum margin among positive tokens in the support token set, $\mathcal{S}$, with an average error measured in this error set, $\mathcal{E}$:

$$\tilde{\Lambda} = \arg\max_{\Lambda} \left[\min_{X_i \in \mathcal{S}} d(X_i) \; - \; \epsilon \cdot \frac{1}{|\mathcal{E}|} \sum_{X_i \in \mathcal{E}} \xi_i(X_i) \right] \tag{6}$$

where $\epsilon > 0$ is a pre-set positive constant to balance contribution from the minimum margin and the average error. For simplicity, the error function $\xi_i(X_i)$ can be directly calculated based on discriminant functions, $\mathcal{F}(\cdot)$. As in multi-class SVM in [9, 1], we have several different ways to define the error function for each negative token X_i:

$$\xi_i(X_i) = \frac{1}{|\Omega|} \sum_{j \in \Omega} \left[\, \mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) \, \right] \tag{7}$$

or

$$\xi_i(X_i) = \max_{j \in \Omega} \left[\, \mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) \, \right] \tag{8}$$

Therefore, the new large margin estimation method is summarized as a constrained maximin optimization problem as follows:

$$\max_{\Lambda} \left[\min_{X_i \in \mathcal{S}} d(X_i) \; - \; \epsilon \cdot \frac{1}{|\mathcal{E}|} \sum_{X_i \in \mathcal{E}} \xi_i(X_i) \right] \tag{9}$$

$$\text{subject to } \; R(\Lambda) = \sum_i \mathcal{D}(\lambda_i \mid\mid \lambda_i^{(0)}) \leq r^2 \tag{10}$$

where r is a pre-set constant. This new method is named as *soft Large Margin Estimation* (*soft-LME* for short). For convenience, we use eq.(7) to calculate the error functions, ξ_i, in this paper.

4. SOLVING SOFT-LME VIA SEMIDEFINITE PROGRAMMING

Obviously, the above constrained optimization in eq.(9) problem can be solved with various optimization methods, such as gradient descent method in [5, 4] and semi-definite programming (SDP) method in [7, 6]. As shown in [7, 6], the SDP method is demonstrated to be superior in both theoretical formulation and practical performance. Thus, in this paper, we consider to convert the above *soft-LME* problem in eqs.(9) and (10) into an SDP problem by following the similar idea used in [7] so that many existing efficient SDP optimization algorithms or tools can be used to solve the *soft-LME* problem as well.

At first, we assume each speech unit is modeled by an N-state CDHMM with parameter vector $\lambda = (\pi, A, \theta)$, where π is the initial state distribution, $A = \{a_{ij} | 1 \leq i, j \leq N\}$ is transition matrix, and θ is parameter vector composed of mixture parameters $\theta_i = \{\omega_{ik}, \mu_{ik}, \Sigma_{ik}\}_{k=1,2,\cdots,K}$ 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 distributions with diagonal covariance matrices:

$$\begin{aligned} p(\mathbf{x}|\theta_i) &= \sum_{k=1}^{K} \omega_{ik} \cdot \mathcal{N}(\mathbf{x} \mid \mu_{ik}, \Sigma_{ik}) \\ &= \sum_{k=1}^{K} \omega_{ik} \prod_{d=1}^{D} \sqrt{\frac{1}{2\pi\sigma_{ikd}^2}} e^{-\frac{(x_d - \mu_{ikd})^2}{2\sigma_{ikd}^2}} \end{aligned} \tag{11}$$

where mixture weights ω_{ik}'s satisfy the constraint $\sum_{k=1}^{K} \omega_{ik} = 1$ and $\Sigma_{ik} = \text{diag}(\sigma_{ik1}^2, \sigma_{ik2}^2, \cdots, \sigma_{ikD}^2)$ denotes the diagonal covarance matrix of k-th Gaussian in state i. For simplicity, we only consider to estimate mean vectors with the soft-LME method.

Next, as in [7], we introduce a new variable $-\rho$ ($\rho > 0$) as the common upper bound for all terms in $d(X_i)$ so that we can convert the maximin optimization in eq.(9) into an equivalent minimization problem as follows:

Problem 1

$$\tilde{\Lambda} = \arg\min_{\Lambda,\rho} \left[-\rho + \frac{\epsilon}{|\mathcal{E}|} \sum_{X_i \in \mathcal{E}} \xi_i(X_i) \right] \quad (12)$$

subject to

$$\mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) \leq -\rho \quad (13)$$

$$R(\Lambda) = \sum_{k=1}^{L} \sum_{d=1}^{D} \frac{(\mu_{kd} - \mu_{kd}^{(0)})^2}{\sigma_{kd}^2} \leq r^2 \quad (14)$$

$$\rho \geq 0. \quad (15)$$

for all $X_i \in \mathcal{S}$ and $j \in \Omega$ and $j \neq W_i$. Since we estimate only Gaussian means, the constraint in eq.(10) can be simplified into eq.(14), where $\mu_{kd}^{(0)}$ represents the original value of μ_{id} in the initial model set.

Then, we introduce some notations to represent **Problem 1** in matrix form: e_i is an L-dimensional vector with -1 at the i-th position, and zero everywhere else. A column vector x is written as $x = (x_1; x_2; \ldots; x_n)$ and a row vector as $x = (x_1, x_2, \ldots, x_n)$. I_D is a $D \times D$ identity matrix. And U is a matrix created by concatenating all normalized Gaussian mean vectors as its columns as:

$$U = (\tilde{\mu}_1, \tilde{\mu}_2, \ldots, \tilde{\mu}_L) \quad (16)$$

where each column is a normalized Gaussian mean

$$\tilde{\mu}_k \ := \ (\mu_{k1}/\sigma_{k1};\ \mu_{k2}/\sigma_{k2};\ \ldots;\ \mu_{kD}/\sigma_{kD}). \quad (17)$$

As shown in [7, 6], the discriminative function, $\mathcal{F}(X|\lambda_W)$ can be represented as the following matrix format under the Viterbi approximation:

$$\mathcal{F}(X|\lambda_W) \approx -A \cdot Z + c \quad (18)$$

where c is a constant irrelevant to HMM means and

$$A = \frac{1}{2} \sum_{t=1}^{T} (\tilde{x}_t; e_{k_t})(\tilde{x}_t; e_{k_t})^T \quad (19)$$

$$Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \qquad Y = U^T U \quad (20)$$

and $\mathbf{k} = \{k_1, k_2, \cdots, k_R\}$ denotes the optimal Viterbi path when evaluating X against model λ_W.

Based on the above notations, we can convert the constraints in eqs.(13) and (14) into the following matrix format [7]:

$$\mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i}) = A_{ij} \cdot Z - c_{ij} \leq -\rho \quad (21)$$

where $A_{ij} = A_i - A_j$ with A_i and A_j calculated according to eq.(19) based on the Viterbi decoding paths **i** and **j** respectively. And

$$R(\Lambda) = Q \cdot Z \leq r^2 \quad (22)$$

where $Q = \sum_{k=1}^{n} (\tilde{\mu}_k^{(0)}; e_k)(\tilde{\mu}_k^{(0)}; e_k)^T$.

Following the same idea, we can also represent the average error in eq.(12) as a matrix form of Z:

$$\begin{aligned} \frac{1}{|\mathcal{E}|} \sum_{X_i \in \mathcal{E}} \xi_i(X_i) &= \frac{1}{|\mathcal{E}||\Omega|} \sum_{X_i \in \mathcal{E}} \sum_{j \in \Omega} [\mathcal{F}(X_i|\lambda_j) - \mathcal{F}(X_i|\lambda_{W_i})] \\ &= \frac{1}{|\mathcal{E}||\Omega|} \sum_{X_i \in \mathcal{E}} \sum_{j \in \Omega} \left[-A'_i \cdot Z + c'_i + A'_j \cdot Z - c'_j \right] \\ &= E \cdot Z + c' \end{aligned} \quad (23)$$

where $E = \frac{1}{|\mathcal{E}||\Omega|} \sum_{X_i \in \mathcal{E}} \sum_{j \in \Omega} (A'_j - A'_i)$ and $c' = \frac{1}{|\mathcal{E}||\Omega|} \sum_{X_i \in \mathcal{E}} \sum_{j \in \Omega} (c'_i - c'_j)$.

Therefore, we have the following minimization problem:

Problem 2

$$\tilde{\Lambda} = \arg\min_{\Lambda,\rho} \left[-\rho + \epsilon \cdot E \cdot Z \right] \quad (24)$$

subject to

$$A_{ij} \cdot Z + \rho \leq c_{ij} \quad \rho \geq 0 \quad (25)$$

$$Q \cdot Z \leq r^2 \quad (26)$$

$$Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \qquad Y = U^T U \quad . \quad (27)$$

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

At last, as in [7], we relax the non-convex constraint $Y = U^T U$ into $Y - U^T U \succeq 0$ to convert **Problem 2** into an SDP problem:

Problem 3

$$\min_{Z,\rho} \left[-\rho + \epsilon \cdot E \cdot Z \right] \quad (28)$$

subject to:

$$A_{ij} \cdot Z + \rho \leq c_{ij} \quad (29)$$

$$Q \cdot Z \leq r^2 \quad (30)$$

$$Z = \begin{pmatrix} I_D & U \\ U^T & Y \end{pmatrix} \succeq 0 \qquad \rho \geq 0 \quad (31)$$

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

Problem 3 is a standard SDP problem, which can be solved efficiently by many SDP algorithms. In problem 3, the optimization is carried out w.r.t. Z (which is constructed from all HMM Gaussian means) and ρ while A_{ij} and c_{ij} and Q and E are constant symmetric matrices calculated from training data, and r and ϵ are two pre-set control parameters. After the solution Z^* to problem 3 is found by an SDP solver, the new HMM means can be obtained from Z^* based on a simple projection (see [7, 6] for details).

5. EXPERIMENTS

The proposed soft-LME/SDP method has been evaluated for a continuous speech recognition task using the TIDIGITS database. Only adult portion of the corpus is used in our experiments. It contains a total of 225 speakers (111 men and 114 women), 112 of which (55 men and 57 women) are used for training and 113 (56 men, 57 women) for test. The training set has 8623 digit strings and the test set has 8700 strings. Our model set consists of 11 whole-word CDHMMs representing all digits. Each HMM has 12 states and use 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).

In our experiments, we first train models based on maximum likelihood (ML) criterion. Next, MCE training uses the best ML model as the seed model. Then, we re-estimate the models with the original LME/SDP method [5] and the proposed soft-LME/SDP, where we use the best MCE models as the initial models and only HMM mean vectors are re-estimated. In each iteration, a number of competing string-level models are computed for each training utterance 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 SDP optimization , i.e. **Problem 3**, is solved with respect to Z and ρ. *In the original LME/SDP method, all training errors are discarded in each iteration of training while all training errors are included in the soft-LME/SDP method to compute matrix E in eq.(28).* In this work, **Problem 3** is solved by an open software, *DSDP v5.6* [3], running under Matlab. At last, CDHMM means are updated based on the optimization solution Z^* found by *DSDP*. If not convergent, next iteration starts again from recognizing all training data to generate N-Best competing strings.

In this work, we have investigated various model sizes, from 1 mixture to 32 mixtures per state. However, for large models (8-mix, 16-mix and 32-mix), training error rate can be quickly reduced to around 0.2% (less than 20 negative tokens), as shown in Table 1. Therefore, the soft-LME/SDP method does not produce any improvement over the original LME/SDP. For small models, especially 1-mix, the soft-LME/SDP yields modest improvements, as shown in Table 2. For example, for 1-mix models, soft-LME/SDP improves string error rate to 2.56% from 2.75% of LME/SDP. More importantly, the soft-LME/SDP converges much faster than the original LME/SDP method since it can effectively use more data in training, including both support tokens and negative tokens, as shown in Figure 1, where we plot learning curves of both LME/SDP and soft-LME/SDP for 1-mix models during the first 12 iterations. Moreover, in the TIDIGITS database, we have observed that we usually need to significantly reduce weight ϵ in eq.(28) as the soft-LME/SDP training proceeds since the soft-LME/SDP can quickly bring down training errors to a very low level in this database. Otherwise, the soft-LME/SDP training is biased by a very small error set, which may contain outliers.

Currently we are testing the soft-LME/SDP method in other more challenging tasks, such as Switchboard and SPINE, where training error rates remain quite high even after discriminative training. We expect the soft-LME/SDP method will make a big difference in these tasks.

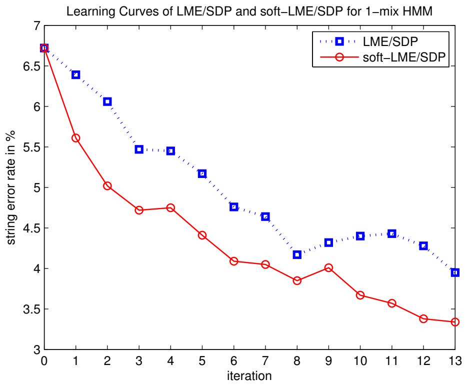

Fig. 1. Learning Curves of LME/SDP and soft-LME/SDP for 1-mix CDHMM in the TIDIGITS test set.

6. CONCLUSIONS

In this work, we studied to incorporate training errors for large margin estimation of CDHMMs under the semi-definite programming framework. Following the idea of soft-margin SVM, we proposed a new soft-LME method, in which training errors can nicely included in large margin training process without significantly increasing complexity of the algorithm. The method has achieved modest performance improvement in TIDIGITS database. More importantly, it yields much faster convergence speed in training which could greatly reduce total training time.

Table 1. String error rates (in %) on the TIDIGITS training data. (ML: maximum likelihood; MCE: minimum classification error; LME: the original LME/SDP in [7]; soft-LME: the proposed soft-LME/SDP.)

	ML	MCE	LME	soft-LME
1-mix	10.80	5.49	1.70	1.33
2-mix	4.01	3.18	0.51	0.42
4-mix	2.11	1.61	0.45	0.30
8-mix	1.07	0.96	0.39	0.26
16-mix	0.78	0.56	0.24	0.24
32-mix	0.56	0.30	0.20	0.20

Table 2. String error rates (in %) on the TIDIGITS test set.

	ML	MCE	LME	soft-LME
1-mix	12.61	6.72	2.75	**2.56**
2-mix	5.26	3.94	1.24	**1.24**
4-mix	3.48	2.23	0.89	**0.87**
8-mix	1.94	1.41	0.68	**0.68**

7. REFERENCES

[1] J. Arenas-Garcia and F. Perez-Cruz, "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, 2003.

[2] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, 2, 121-167 (1998).

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

[4] 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.

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

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

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

[8] C. Liu, H. Jiang and L. Rigazio, "Recent Improvements on Maximum Relative Margin Estimation of HMMs for Speech Recognition," *Proc. of 2006 IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP2006)*, Toulouse, France, May 2006.

[9] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," *Proc. of European Symposium on Artificial Neural Networks*, 1999.