Soft Margin Estimation with Various Separation Levels for LVCSR

Jinyu Li[1], *Zhi-Jie Yan*[2], *Chin-Hui Lee*[1] *and Ren-Hua Wang*[2]

[1]School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA, 30332
[2]iFlytek Speech Lab, University of Science and Technology of China, Hefei, P. R. China, 230027
jinyuli@ece.gatech.edu yanzhijie@ustc.edu chl@ece.gatech.edu rhw@ustc.edu.cn

ABSTRACT

We continue our previous work on soft margin estimation (SME) to large vocabulary continuous speech recognition (LVCSR) in two new aspects. The first is to formulate SME with different unit separation. SME methods focusing on string-, word-, and phone-level separation are defined. The second is to compare SME with all the popular conventional discriminative training (DT) methods, including maximum mutual information estimation (MMIE), minimum classification error (MCE), and minimum word/phone error (MWE/MPE). Tested on the 5k-word Wall Street Journal task, all the SME methods achieves a relative word error rate (WER) reduction from 17% to 25% over our baseline. Among them, phone-level SME obtains the best performance. Its performance is slightly better than that of MPE, and much better than those of other conventional DT methods. With the comprehensive comparison with conventional DT methods, SME demonstrates its success on LVCSR tasks.

Index Terms: soft margin estimation, hidden Markov model, discriminative training

1. INTRODUCTION

Discriminative training (DT) methods have been extensively studied to boost the automatic speech recognition (ASR) system accuracy [1-3]. The most successful methods are maximum mutual information estimation (MMIE) [1], minimum classification error (MCE) [2], and minimum word/phone error (MWE/MPE) [3]. MWE/MPE has achieved great successes by minimizing the approximate word/phone error rates, compared to MCE and MMIE that minimize some utterance-level error measures. Attributed to the success of MWE/MPE, several variations have been proposed recently, such as minimum divergence training [4] and minimum phone frame error training [5].

Inspired by the great success of margin-based classifiers, there is a trend to incorporate the margin concept into hidden Markov model (HMM) for ASR. In contrast to the above conventional DT methods, margin-based techniques treat the generalization issue from a perspective of statistical learning theory [6]. Several attempts based on margin maximization were proposed recently and have shown some advantages over DT methods in some small ASR tasks. Major methods are: large margin estimation (LME) [7], large margin hidden Markov models (LM-HMMs) [8], and soft margin estimation (SME) [9].

Although margin-based methods have shown their superiority on small tasks, they have not well demonstrated the same power on large vocabulary continuous speech recognition (LVCSR) tasks. To be widely used, it is necessary to show convincing successes on LVCSR tasks. In [10], SME was shown to work well on the 5k-word Wall Street Journal (5k-WSJ0) [11] task. However, two potential issues need to be addressed. The first is that the criterion to judge correct and competing candidates was on the string level in [10], although frame-based selection was performed. If word- or phone-level criterion can be used, performance may be boosted according to the success of MWE/MPE. The second is that SME only showed advantage over MCE in [10] and didn't compare with MMIE and MPE on the same task. The newest version of HMM toolkit (HTK) [12] provides an accurate tool to train MMIE and MWE/MPE models. We should test the performance of MMIE and MWE/MPE models trained by HTK.

This study addresses the two above-mentioned issues. For the unit separation, string, word, and phone levels are all considered. Different separation measures are defined based on the candidate definition. For comparison, MCE will be compared fairly with SME by sharing most of implementation details. In addition, MMIE, MWE, and MPE trained by HTK will also be compared.

The research presented in this paper will make a comprehensive study of SME for LVCSR tasks. In Section 2, we propose SME with utterance-, word-, and phone-level unit separation. In Section 3, SME will be compared with all the popular DT methods. The best SME model achieves a relative word error rate (WER) reduction of 25% from our maximum likelihood estimation (MLE) baseline on the 5k-WSJ0 task. It outperforms all the DT models in this study. Conclusions are drawn in Section 4.

2. SOFT MARGIN ESTIMATION

In this section, the theory of soft margin estimation is first briefly reviewed. A frame-based SME framework is proposed. Then SME methods with different unit separation are formulated. Finally, the practical implementation issues are discussed.

2.1 Previous SME Work

Here, we briefly introduce SME. Please refer to [9] for detailed discussion. SME has two targets for optimization: one is to minimize the empirical risk, and the other is to maximize the margin. These two targets are combined into a single SME objective function for minimization:

$$L^{SME}(\Lambda) = \frac{\lambda}{\rho} + \frac{1}{N}\sum_{i=1}^{N} \ell\left(O_i, \Lambda\right). \quad (1)$$

Λ denotes the set of HMM parameters, $\ell\left(O_i, \Lambda\right)$ is a loss function for utterance O_i, and N is the number of training utterances. ρ is a constant soft margin, and λ is a coefficient to balance soft margin maximization and empirical risk minimization.

As discussed in [9], there is a mapping relationship between λ and ρ. For a fixed λ, there is one corresponding ρ. Instead of choosing a fixed λ and trying to get the solution of Eq. (1), we can directly choose a ρ in advance. The key component of SME is a proper definition of the loss function, $\ell(O_i,\Lambda)$. This loss should be related to the margin, ρ.

In [10], we have elaborated to design the loss function $\ell(O_i,\Lambda)$. Different loss functions for SME with frame or utterance selection have been discussed. Frame-based SME with a modified hinge loss function was shown to outperform other options. In this study, we will directly use that loss function. For SME, our target is to separate the correct candidates from the competing candidates with a distance greater than the value of the margin. How to define the candidates is an issue. In [10], the candidates are defined in the string level. However, this is different from the target of ASR, which wants to reduce the word error rate instead of the string error rate. In this paper, we further works on the word- and phone-level separation for SME. This turns out to be very effective to reduce the word error rate on LVCSR tasks.

2.2 SME with Frame Selection

SME with frame selection is formularized as:

$$\ell(O_i,\Lambda)=\sum_j \ell(O_{ij},\Lambda)=\sum_j \{(\rho-d(O_{ij},\Lambda))I(O_{ij}\in F_i)\}, \quad (2)$$

where O_{ij} is the jth frame for utterance O_i, $I(.)$ is an indicator function, and F_i is the frame set in which the frames contribute to the loss computation. $d(O_{ij},\Lambda)$is the separation measure between the correct and competing candidates for O_{ij}. SME now selects the frames that are critical to discriminative separation. We realize it with the frame posterior probability via computing the posterior probability $p(c\mid t_{cs},t_{ce},O_i)$ for a correct candidate c with starting time t_{cs} and ending time t_{ce}. The candidate can be word or phone as described in the following section. The frame posterior probability is then computed by summing the posterior probabilities of all the correct candidates that pass time j:

$$q(O_{ij})=\sum_{c\mid t_{cs}<=j<=t_{ce}} p(c\mid t_{cs},t_{ce},O_i). \quad (3)$$

Frame selection for SME is done by comparing the frame posterior probability with a margin ρ and a threshold τ:

$$\ell(O_{ij},\Lambda)=\begin{cases}\rho-d(O_{ij},\Lambda), & \text{if } \rho>q(O_{ij})>\tau\\ 0, & \text{otherwise}\end{cases}. \quad (4)$$

Eq. (4) selects the frames that are critical for parameter updating. As discussed in [10], it works on confusion patterns and removes the influence of noisy frames with too small posterior probabilities because they may be unreliable for parameter update due to inacurate time alignment. This loss function has been demonstrated to have the best performance in [10]. Please refer [10] for the comparison of different loss functions.

2.3 SME with Unit Separation at Different Levels

The computation of frame probability $q(O_{ij})$ and separation measure $d(O_{ij},\Lambda)$ rely on how to define the correct and competing candidate units. This will be discussed in the following section with string-, word-, and phone-level candidate unit definitions.

2.3.1 String-level Unit

In this case, SME separates the correct strings from incorrect strings in a decoded lattice. For every word w in the decoded lattice, it is considered as a correct word only if it belongs to a lattice path, which corresponds to the correct transcription S_i for utterance O_i. Then the posterior probability for a correct word w with starting time t_{ws} and ending time t_{we} is got by summing the probabilities of all the lattice paths, R, in which w lies in:

$$p(w\mid t_{ws},t_{we},O_i)=\sum_{\substack{R\in G_i\wedge(w\mid t_{ws},t_{we})\in R\\ \wedge(w\mid t_{ws},t_{we})\in S_i}}\frac{P_\Lambda(O_i\mid R)P(R)}{\sum_{\hat{S}\in G_i}P_\Lambda(O_i\mid\hat{S})P(\hat{S})}, \quad (5)$$

where G_i is a decoded lattice for utterance O_i, and $\hat{S}$ denotes the transcription of words in the decoded lattice, G_i.

The frame posterior probability is then computed by summing the posterior probabilities of all the correct words passing time j:

$$\begin{aligned}q(O_{ij})&=\sum_{w\mid t_{ws}<=j<=t_{we}} p(w\mid t_{ws},t_{we},O_i)\\ &=\sum_{\substack{w\mid\\ t_{ws}<=j<=t_{we}}}\sum_{\substack{R\in G_i\wedge(w\mid t_{ws},t_{we})\in R\\ \wedge(w\mid t_{ws},t_{we})\in S_i}}\frac{P_\Lambda(O_i\mid R)P(R)}{\sum_{\hat{S}\in G_i}P_\Lambda(O_i\mid\hat{S})P(\hat{S})}.\end{aligned} \quad (6)$$

The last step is to define a frame level separation measure as:

$$d(O_{ij},\Lambda)=\log\sum_{\substack{w\mid\\ t_{ws}<=j<=t_{we}}}\sum_{\substack{R\in G_i\wedge(w\mid t_{ws},t_{we})\in R\\ \wedge(w\mid t_{ws},t_{we})\in S_i}}\frac{P_\Lambda(O_i\mid R)P(R)}{\sum_{\hat{S}\in G_i\wedge\hat{S}\neq S_i}P_\Lambda(O_i\mid\hat{S})P(\hat{S})}. \quad (7)$$

The correct transcription is removed from the denominator in Eq. (7) because it is a measure of the correct versus the incorrect transcriptions. By plugging Eqs. (6) and (7) into Eq. (4) to compute the loss functions in Eq. (1), SME with string-level unit separation is implemented.

2.3.2 Word-level Unit

In this case, we want to separate the correct words from the incorrect words in a decoded lattice. For the correct transcription S_i that consists of N_i correct words, its correct words set is

$$W_i=\{w_i^1,w_i^2,.....w_i^{N_i}\}. \quad (8)$$

Every word in set W_i has a unique label, a starting and an ending time. For any word w in set W_i with a time interval [t_{ws}, t_{we}], we look at all the strings in the decoded lattice. If a segment has the same label with w and also spans [t_{ws}, t_{we}], we consider the strings passing that segment as positive candidates and separate these positive strings from other strings in the duration [t_{ws}, t_{we}]. In this way, we can directly work on the local word level of the decoded lattice, and maximize the word accuracy.

To formulate the counter parts of Eqs. (5), (6), and (7) with word-level unit, we need to define two string sets for each word w_i^n in set W_i. One set (Φ_i^n) contains all the strings passing w_i^n in the decoded lattice, G_i:

$$\forall R\in\Phi_i^n,\exists w\in R\wedge w=w_i^n. \quad (9)$$

The other set ($\hat{\Phi}_i^n$) contains all the strings that do not pass w_i^n:

$$\forall\hat{S}\in\hat{\Phi}_i^n,\forall\hat{w}\in\hat{S}\wedge\hat{w}\neq w_i^n. \quad (10)$$

In contrast to the string-level operation, a word w in the decoded lattice is considered to be a correct word when it is in set Φ_i^n (i.e., it has the same word label, starting and ending time as w_i). The

posterior probability of word w is now defined by summing the probabilities of all the paths in set Φ_i^n that cross word w:

$$p(w \mid t_{ws}, t_{we}, O_i) = \sum_{R \in \Phi_i^n \wedge (w \mid t_{ws}, t_{we}) \in R} \frac{P_\Lambda(O_i \mid R) P(R)}{\sum_{\hat{S} \in G_i} P_\Lambda(O_i \mid \hat{S}) P(\hat{S})} \quad (11)$$

The frame posterior probability is again computed by summing the posterior probabilities of all the correct words that pass time j:

$$q(O_{ij}) = \sum_{w \mid t_{ws} <= j <= t_{we}} p(w \mid t_{ws}, t_{we}, O_i) \quad (12)$$

The frame level separation measure is defined to separate the strings cross time j in set Φ_i^n from the strings in set $\hat{\Phi}_i^n$:

$$d(O_{ij}, \Lambda) = \log \sum_{\substack{w \mid \\ t_{ws} <= j <= t_{we}}} \sum_{R \in \Phi_i^n \wedge (w \mid t_{ws}, t_{we}) \in R} \frac{P_\Lambda(O_i \mid R) P(R)}{\sum_{\hat{S} \in \hat{\Phi}_i^n} P_\Lambda(O_i \mid \hat{S}) P(\hat{S})} \quad (13)$$

By plugging Eqs. (12) and (13) into Eq. (4) to compute the loss functions in Eq. (1), SME with word-level separation is realized.

2.3.3 Phone-level Unit

An extension from word- to phone-level unit is straightforward. We replace all the word denotations from Eq. (8) to Eq. (13) with phone denotations.

2.4 Practical Implementation Issues

The extended Baum-Welch (EBW) [13][14] is adopted to update HMM parameters for all the SME formulations. A brief implementation is described as follows. First, an MLE model and a bigram language model (LM) were used to decode all training utterances to generate corresponding word lattices. Then a unigram was used to rescore the decoded lattices. In all the DT methods experimented in this study, a factor of 1/15 was used to scale down the acoustic model likelihood as used in the other DT studies [3][14][15]. Updating statistics were obtained from the lattices with a forward-backward algorithm. Then, EBW was used to update the HMM parameters as in [14]. Because SME directly works on generalization, no i-smoothing was used.

The word or phone posterior probability computation is a key to implement the EBW algorithm. For the denominator lattice in EBW, we need to remove all the correct candidates and rescale the remaining word or phone arcs to get the posterior probabilities. For SME with string-level candidate unit, any word in a correct string may be shared with an incorrect string. Therefore, we need to consider all the word arcs and follow the method proposed in [15] to compute the word posterior probability:

$$\gamma(w \mid t_{ws}, t_{we}, O_i) = \frac{\sum_{R \in G_i \wedge (w \mid t_{ws}, t_{we}) \in R} P_\Lambda(O_i \mid R) P(R) - \sum_{\substack{R \in G_i \wedge R = S_i \\ \wedge (w \mid t_{ws}, t_{we}) \in R}} P_\Lambda(O_i \mid R) P(R)}{\sum_{\hat{S} \in G_i} P_\Lambda(O_i \mid \hat{S}) P(\hat{S}) - \sum_{\hat{S} \in G_i \wedge \hat{S} = S_i} P_\Lambda(O_i \mid \hat{S}) P(\hat{S})} \quad (14)$$

It is noted that the posterior probability in Eq. (14) is for the decoded lattice after the correct strings removed. This is different from the posterior probability in Eq. (5), which is for the original decoded lattice.

For SME with word- and phone-level units, it is much simpler. We just consider all the strings passing the incorrect words in a time interval [t_{ws}, t_{we}] and compute the posterior probability as:

$$\gamma(w \mid t_{ws}, t_{we}, O_i) = \sum_{\hat{S} \in \hat{\Phi}_i^n \wedge (w \mid t_{ws}, t_{we}) \in \hat{S}} \frac{P_\Lambda(O_i \mid \hat{S}) P(\hat{S})}{\sum_{\hat{S} \in \hat{\Phi}_i^n} P_\Lambda(O_i \mid \hat{S}) P(\hat{S})} . \quad (15)$$

However, the rescaling of Eq. (15) is aggressive since it will magnify the contribution from segments that have large correct word posterior probabilities. For example, consider a segment has two wrong words. Each word has an original posterior probability of 0.05. All other words in that segment are correct words. After rescaling, those two words will each have posterior probability of 0.5, 10 times as their original probabilities. This magnification will hurt the optimization. Therefore, we set a threshold β and only rescale the segment that has a correct word posterior probability less than β.

3. EXPERIMENTS

The 5k-WSJ0 task was used to evaluate the effectiveness of DT methods on LVCSR. The training set is the SI-84 set, with 7077 utterances from 84 speakers. All testing is conducted on the Nov92 evaluation set, with 330 utterances from 8 speakers. Baseline HMMs are trained with MLE using HTK [12]. The HMMs are cross-word triphone models. There were 2818 shared states obtained with a decision tree and each state observation density is modeled by an 8-mixture Gaussian mixture model. The input features were 12MFCCs + energy, and their first and second order time derivatives. A trigram LM within the 5k-WSJ0 corpus was used for decoding. The baseline WER was 5.06% for MLE models.

The MMIE, MWE, and MPE models were trained with HTK. The i-smoothing factor was set 100 for MMIE, 25 for MWE, and 50 for MPE, as suggested in [14]. The word error rates (WER) of MMIE, MWE, and MPE on the Nov92 evaluation set are 4.60%, 4.37%, and 3.92%, which correspond to 9%, 14%, and 22% relative WER reductions, respectively.

The MCE model was also trained. The smoothing constant in the sigmoid function was set to 0.04 as in [15]. EBW was used for HMM parameters update. The WER of the MCE model was 4.60%, getting 9% relative WER reduction over the MLE baseline.

For the purpose of a fair comparison, all the proposed SME methods were modified on the basis of the MCE implementation. This means that the implementations are similar, only the individual algorithm parts are different. Three SME models were trained with the string-, word-, and phone-level units separation. They are denoted as SME_String, SME_Word, and SME_Phone, separately. All the SME models were initiated from MLE model. For all the SME models training, the margin ρ and the threshold τ were set as 0.9 and 0.1, individually. For SME_Word and SME_Phone, the threshold β was set as 0.4 for rescaling the posterior probabilities of incorrect arcs in the decoded lattice.

Table 1: Performance comparison on the 5k-WSJ0 task.

	WER	Relative Improvement
MLE	5.06%	-
MCE	4.60%	9%
MMIE	4.60%	9%
MWE	4.37%	14%
MPE	3.92%	22%
SME_String	4.22%	17%
SME_Word	4.11%	19%
SME_Phone	**3.81%**	**25%**

Table 1 compares the resulting WERs and relative WER reductions of the conventional DT methods (MCE, MMIE, MWE, and MPE) and all the SME methods (SME_String, SME_Word,

and SME_Phone) from the MLE baseline. The conventional DT methods got 9%-22% relative WER reductions. Among them, MPE performed the best, obtaining 22% relative WER reductions. All the proposed SME methods worked better than most of the conventional DT methods, achieving about 17%-25% relative WER reduction from MLE baseline. SME_Phone is the best among all the evaluated methods, reaching 25% relative WER reductions. This shows the objective of phone-level separation is very effective on the 5k-WSJ0 task. Working on the string-level discrimination, SME_String is better than MCE and MMIE. Focusing on the word level, SME_Word also outperforms MWE. The relation is also true for SME_Phone and MPE.

The evolutions of WERs of the best two DT methods, MPE and SME_Phone, are plotted in Figure 1. The minimum WERs of MPE and SME_Phone were reached at iteration 12. All the other methods also reached their minimum WERs within 15 iterations. To save space, we didn't plot them. It is almost in every iteration that the WER of SME_Phone was less than that of MPE.

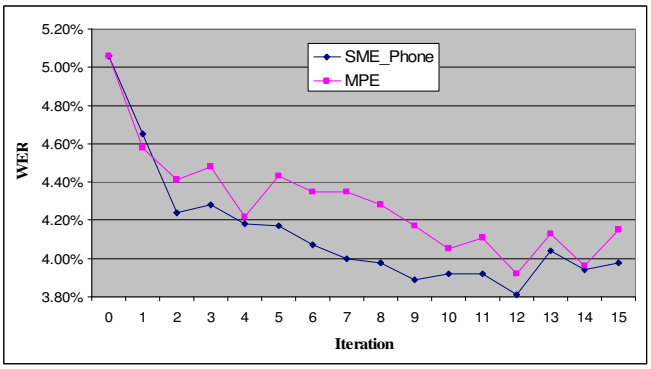

Figure 1: Evolutions of testing WER for MPE and SME_Phone models on the 5k-WSJ0 task.

4. CONCLUSIONS

In this study, we treated SME thoroughly on the 5k-WSJ0 task. SME methods for string-level (SME_String), word-level (SME_Word), and phone-level (SME_Phone) separation are proposed. All these methods focus on how to make the correct units separate from the competing units with a distance greater than the value of the margin. The most popular DT methods (MMIE, MCE, MWE, and MPE) are used for comparison on the 5k-WSJ0 task. In each separation level, SME outperforms its counter part of conventional DT methods. In string level, SME_String gets 17% relative WER reductions while MCE and MMIE all get 9% relative WER reductions. In word level, SME_Word obtains 19% relative WER reductions while MWE has 14%. In phone level, SME_Phone achieves 25% relative WER reductions, compared to 22% for MPE. We can see that phone-level separation optimization is the most effective way to reduce WER on the 5k-WSJ0 task. SME works the best in this study and demonstrates that the margin-based methods can be a good option for LVCSR tasks.

Two research issues need to be addressed in the future. The first is to extend this study to feature extraction on an LVCSR task. fMPE [16] has already demonstrated its great power in LVCSR tasks. SME only showed its success in jointly optimization of features and HMM parameters on the TIDIGITS task [17]. We will work SME on the feature extraction part and compare with fMPE on an LVCSR task. The second is to apply SME to even larger LVCSR tasks than the 5k-WSJ0 task.

5. REFERENCES

[1] Bahl, L. R., Brown, P. F., de Souza, P.V., and Mercer, R. L., "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," *Proc. ICASSP*, vol. 1, pp. 49-52, 1986.

[2] Juang, B. -H., Chou, W., and Lee, C. -H., "Minimum classification error rate methods for speech recognition," *IEEE Trans. on Speech and Audio Proc.*, vol. 5, no. 3, pp. 257-265, 1997.

[3] Povey, D. and Woodland, P., "Minimum phone error and i-smoothing for improved discriminative training," *Proc. ICASSP*, vol. 1, pp. 105-108, 2002.

[4] Du, J., Liu, P., Soong, F. K., Zhou, J.-L., and Wang, R.-H., "Minimum divergence based discriminative training," *Proc. Interspeech*, pp. 2410-2413, 2006.

[5] Zheng J. and Stolcke, A., "Improved discriminative training using phone lattices," *Proc. Interspeech*, pp. 2125-2128, 2005.

[6] Vapnik, V., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

[7] Jiang, H., Li, X., and Liu, C., "Large margin hidden Markov models for speech recognition," *IEEE Trans. On Audio, Speech, and Language Proc.*, vol. 14, pp. 1584-1595, 2006.

[8] Sha, F. and Saul, L. K., "Large margin hidden Markov models for automatic speech recognition," *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J.C. Platt, and T. Hofmann, Eds., MIT Press, 2007.

[9] Li, J., Yuan, M., and Lee, C. -H., "Approximate test risk bound minimization through soft margin estimation," *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 15, no. 8, pp. 2393- 2404, 2007.

[10] Li, J., Yan, Z., Lee, C. -H., and Wang, R. -H., "A study on soft margin estimation for LVCSR," *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 268-271, 2007.

[11] Paul, D. B. and Baker, J. M., "The design for the wall street journal-based CSR corpus," *Proceedings of the workshop on Speech and Natural Language*, 1992.

[12] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. C., *The HTK Book (for HTK Version 3.4)*, Cambridge University, 2006.

[13] Normandin, Y., *Hidden Markov models, maximum mutual information estimation, and the speech recognition problem*. Ph.D. thesis, McGill University, 1991.

[14] Povey, D., *Discriminative training for large vocabulary speech recognition*, Ph.D. thesis, Cambridge University Engineering Dept, 2003.

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

[16] Povey, D., Kingsbury, B., Mangu, L., Saon, G., Soltau, H., Zweig, G., **"**fMPE: discriminatively trained features for speech recognition," *Proc. ICASSP*, pp. 961 - 964, 2005.

[17] Li, J. and Lee, C. -H., "Soft margin feature extraction for automatic speech recognition," *Proc. Interspeech*, 2007.