Phone-Discriminating Minimum Classification Error (P-MCE) Training for Phonetic Recognition

Qiang Fu[1], *Xiaodong He*[2], *Li Deng*[2]

[1]ECE Department, Georgia Institute of Technology, Atlanta, GA 30332, USA
[2]Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA

qfu@ece.gatech.edu, {xiaohe,deng}@microsoft.com

Abstract

In this paper, we report a study on performance comparisons of discriminative training methods for phone recognition using the TIMIT database. We propose a new method of *phone-discriminating minimum classification error (P-MCE)*, which performs MCE training at the sub-string or phone level instead of at the traditional string level. Aiming at minimizing the phone recognition error rate, P-MCE nevertheless takes advantage of the well-known, efficient training routine derived from the conventional string-based MCE, using specially constructed one-best lists selected from phone lattices. Extensive investigations and comparisons are conducted between the P-MCE and other discriminative training methods including maximum mutual information (MMI), minimum phone or word error (MPE/MWE), and the other two MCE methods. The P-MCE outperforms most of experimented approaches on the standard TIMIT database in terms of the continuous phonetic recognition accuracy. P-MCE achieves comparable results with the MPE method which also aims at reducing phone-level recognition errors.

Index Terms: phonetic recognition, minimum classification error, discriminative training

1. Introduction

Discriminative training (DT) has been an active research area in speech recognition for many years. The popular DT methods can be classified into three categories: minimum classification error (MCE) [1, 2, 3], maximum mutual information (MMI) [4], and minimum phone/word error (MPE/MWE) [5]. Remarkable success is reported on both small vocabulary tasks (e.g., TIDIGITS) and large vocabulary tasks (e.g. WSJ). In the past several years, (continuous) phonetic recognition has become attractive in light of recent surge in the interest of new applications such as spoken document indexing/retrieval and spontaneous speech recognition. However, the effects of DT methods on phonetic recognition have not been studied in a systematic manner in the past. Because the phones as the recognition units in phonetic recognition are much smaller than the word units used in continuous speech recognition, it is not clear whether the effectiveness of the DT methods can be transferred directly to phonetic recognition tasks.

In the research reported in this paper, we conduct extensive investigation and comparisons of various DT methods on the standard phonetic recognition task defined in the TIMIT database [6, 7]. We select the MCE method as the core algorithm in this discussion because it elaborates on the most direct connection between the Bayes decision rule and the speech recognition performance (i.e., empirical error rate). In particular, given the nature of the traditional string-level MCE [1] [8] which optimizes the sentence error rate instead of the desired phone error rate, we develop a new version of the MCE method, which is named phone-discriminating MCE (P-MCE).

The MCE method for any specific application can be formulated by the following steps [9]:

1. Define the performance objective and the corresponding task evaluation measure;
2. Specify the target event (i.e., the correct label), competing events (i.e. the incorrect hypotheses resulting from the recognizer), and the corresponding models (as well as the organization of the training events);
3. Construct the objective function and set values of the hyper-parameters;
4. Choose a suitable optimization method to estimate model parameters.

For phonetic recognition, the performance measure is the phone recognition accuracy naturally. The objective function can be written as the empirical expectation of the smoothed error accordingly and the optimization method can be either gradient-based methods such as the generalized probabilistic descent (GPD) [1] method or extended Baum-Welch (EBW) algorithm [10, 3]. While the selection of competing events appears to be trivial theoretically after the classification error is defined, it requires careful considerations in the algorithm implementation, which forms the essence of the issue discussed in this paper. In phonetic recognition tasks, there are two possible MCE schemes to determine the competing events. First, the conventional string-based MCE using N-best lists treats a whole utterance (i.e. a string) as a training "token", where there is no need to specify the phone boundaries explicitly. This leads to the improvement of string, instead of phone, recognition accuracy. In the second scheme, the competing events can be selected using phone lattices, which contain a richer search space and cover more competing candidates than N-best lists. In this case, for any specific phone in the transcription, an ideal scheme of competing events selection for continuous phonetic recognition would be to generate a phone lattice and take the arcs with identical segmentation but different identities. However, the segmentation in phone lattices is neither reliable nor strongly consistent with that for the labeled phones. One compromised solution would be to relax the segmentation-boundary constraint, which we have found often leads to inaccurate phone error calculation.

The first author performed the work while at Microsoft Research as a summer intern in 2006

The weaknesses of both schemes above for selecting the competing events are overcome by the novel technique of P-MCE introduced in this paper. While both P-MCE and MPE [5] focus on phone discrimination, P-MCE does not require a number of heuristics used in MPE in defining the phone recognition accuracy. In essence, P-MCE selects competing events from the phone lattice, forms a large number of one-best lists each containing only a single phone error from the reference, and then carries out the optimization procedure in the same way as in the conventional string-based MCE training. In this way, assignment of phone boundaries are no longer needed, in contrast to MPE which relies on such assignment.

The rest of the paper is organized as follows. We will describe details of the P-MCE method in Section 2. Comparative experimental results among different discriminative training methods are presented in Section 3. And in Section 4, conclusions are drawn and the planned future work are discussed.

2. Phone-Discriminating MCE Technique

We now follow the four-step procedure discussed above to introduce the P-MCE method. Since the performance measure is phone recognition accuracy by definition, we start from the second step.

2.1. Target and competing events

Figure1 illustrates the principle of our novel competing-token selection in the P-MCE method. Assume the target phone sequence (as the labeled transcription provided by the database) is " ABC " and a possible recognized phone sequence is " A'B'C' ". This competing sequence may be the top-one output from the recognizer or be extracted from the phone lattice generated by the recognizer. As the figure shows, our goal is to create a competing utterance which differs in *only one phone* from the transcription. That is, for each phone in the transcribed target utterance, we form a new phone sequence which is identical to the target utterance except for that phone. For instance, the competing phone sequence for the first phone "A" is " A'BC ". Consequently, we can conduct a conventional string-based MCE over "ABC" and " A'BC ". But since the target and competing sequences differ in only one phone ("A" vs. " A' "), the training of model parameters will be focused on the discrimination of phone "A" vs. " A' ", reaching the goal of phone-level discrimination. The major advantage for the above P-MCE method is that the statistics used for updating the underlying phone parameters are not contaminated by other models, and that there is no need for estimating the phone boundary explicitly in order to compute the necessary likelihoods. In the current implementation of the P-MCE technique, we only form one competing phone sequence (for each incorrect phone in the original misrecognized phone sequence) and conduct one-best MCE training. The competing phone " A' " (or " B' " or " C' ") is chosen from a pre-generated phone lattice. If there is no suitable competitor found in the phone lattice, a competitor is selected using a phone confusion matrix generated from the baseline recognizer.

2.2. The objective function and the optimization method

The objective function of the P-MCE is very similar to that of the string-based MCE, which is

$$L_{P-MCE}(\Lambda) = \frac{1}{R'} \sum_{r=1}^{R'} l(d_r(X_r|\Lambda)) \qquad (1)$$

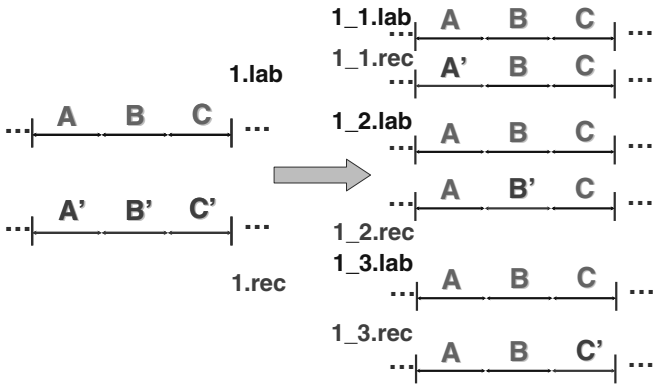

Figure 1: Illustration of creating competing training tokens for P-MCE.

where R' is the total number of training utterances. Note the number of R' is no longer the original number of utterances in the training set. According to the procedure described in the preceding subsection, the total number of P-MCE training tokens is expanded from the original R to $R' = \sum_{r=1}^{R} N_r$, in which N_r is the number of phones in the rth utterance. In Eq.1, $l(.)$ is the sigmoid loss function and $d_r(.)$ is the misclassification measure, both are defined in the same way as in the conventional MCE training [1]. Naturally, the optimization method can be the GPD method or more advanced gradient descent methods such as Quickprop [11].

In this paper, we use a modified EBW algorithm proposed in [3], which gives a solid theoretical basis, stable convergence, and is well suited for the large-scale batch-mode training process essential in large-scale speech recognition and other pattern recognition applications. Here, we provide a brief review for optimizing the objection function of Eq.(1).

For one-best MCE, we define $s_r \in \{S_r, s_{r,1}\}$, where S_r is the correct label sequence for the rth utterance and $s_{r,1}$ is the best competitor (normally the best recognized string not equal to S_r). We can have a misclassification measure function

$$d_r(X_r|\Lambda) = -\log p(X_r, S_r|\Lambda) + \log p(X_r, s_{r,1}|\Lambda) \qquad (2)$$

Note that we use the total likelihood $\log p(X_r|\Lambda)$ instead of the likelihood of the best state sequence $\log p(X_r, q|\Lambda)$ in calculating Eq. (2). Consequently, no Viterbi alignment [1] is needed, in contrast to the conventional [1] which requires Viterbi alignment before computing gradient. This eliminates any problem arising from the inaccurate phone segmentation.

We now define

$$\gamma_{m,r,s_r}(t) = p(q_{r,t} = m|X_r, s_r, \Lambda') \qquad (3)$$

as the posterior probability of being in state m in the corresponding HMM at time t given utterance r for word string s_r. In Eq. (3), q is the state sequence and Λ' is the HMM parameter set in the previous iteration, and $\gamma_{m,r,s_r}(t)$ is computed by the standard forward-backward algorithm.

We further define

$$\begin{aligned} \Delta\gamma_{m,r}(t) \quad &= \quad p(S_r|X_r, \Lambda')p(s_{r,1}|X_r, \Lambda') \cdot \\ &\quad (\gamma_{m,r,S_r}(t) - \gamma_{m,r,s_{r,1}}(t)) \end{aligned} \qquad (4)$$

According to [3], the parameter updating equations are as

following:

$$\mu_m = \frac{\sum_r \sum_t \Delta\gamma_{m,r}(t) x_{r,t} + D_m \mu'_m}{\sum_r \sum_t \Delta\gamma_{m,r}(t) + D_m} \quad (5)$$

$$\begin{aligned} \Sigma_m &= \frac{1}{\sum_r \sum_t \Delta\gamma_{m,r}(t) + D_m} \times \\ &\{\sum_r \sum_t [\Delta\gamma_{m,r}(t)(x_{r,t} - \mu_m)(x_{r,t} - \mu_m)^T] \\ &+ D_m \Sigma'_m + D_m(\mu_m - \mu'_m)(\mu_m - \mu'_m)^T\} \end{aligned} \quad (6)$$

where μ'_m and Σ'_m are the HMM parameters from the previous iteration of the algorithm. In Eq. (6), D_m takes the following functional form:

$$\begin{aligned} D_m &= E \cdot \sum_{r=1}^{R'} p(S_r|X_r, \Lambda')[p(S_r|X_r.\Lambda') \sum_t \gamma_{m,r,S_r}(t) \\ &+ p(s_{r,1}|X_r.\Lambda') \sum_t \gamma_{m,r,s_{r,1}}(t)] \end{aligned} \quad (7)$$

where E is set to be 2 in all the experiments.

2.3. N-best vs. lattice in selecting competing tokens

One critical issue related to the P-MCE implementation is the selection of competing events using N-best lists versus using phone/word lattices. Conventionally, we adopted the phone lattice or lattice approach for two reasons. First, "N-best lists" are at the level of competing *utterances*, from which it is difficult to identify the desired phone/word competitors. Second, lattices contain a richer search space than N-best list. One weakness of the phonetic lattice approach, however, is the often unreliable arc segmentation in the lattice. Another weakness is the possibility of having a sub-string with two or more phones are so acoustically cohesive that they should be adjusted as a single unit. These problems are particularly severe for phone recognition because the duration of phones are much shorter than that of words and hence the phone boundary inaccuracy will have a direct negative impact on the quality of the selected competing events.

For the P-MCE training algorithm, however, all of the above problems are reduced. The algorithm performs discriminative training on an arbitrary sub-string, as long as an appropriate competing utterance can be selected. (We showed the selection of phone-level competing "utterances" at the beginning of this section.) For full exploration of the rich search space afforded by the lattice, we can create N-best lists from the lattice instead of the one-best lists as we have currently implemented the P-MCE algorithm. With a large value of N for selecting competing phones or substrings from a phone lattice, the richness of the search space in the phone lattice would not be wasted as in the current simplistic implementation. How to optimally organize the competing events in the P-MCE framework is a future research direction.

3. Experiments and Results

All experiments reported in this section are carried out on the TIMIT database and we use the standard experimental setup as specified in [6, 7]. We experimented with and compared the performance of various discriminative training techniques including the conventional string-based MCE, the P-MCE, the MMI, and the MPE methods. The phone lattice in the training are generated using the tool of HVite in the HTK toolkit (http://htk.eng.cam.ac.uk/) by setting $n = 3$ (maximum 3 tokens at one frame) $p = 0$ (no phone insertion penalty), and $s = 8$ (language model scale factor).

3.1. Baseline system

A high-quality baseline system is built using the HTK, with carefully constructed decision trees to establish triphone HMMs using bigram. A total of 48 monophone units used follow the exact definition as in [6, 7]. Triphone HMMs are built by maximum likelihood training using 39 MFCC feature vectors (12MFCC +12Δ+12$\Delta\Delta$+3 log energy values). All models except for the short pause unit "sp" are 3-state left-to-right HMMs. Each state has 16 Gaussians except for "sil" which contains 28 Gaussians. The short pause model "sp" has only one state with 16 Gaussians. There are a total of 224077 logical triphone models and 7496 physical triphone models, with 917 physical states after automatic decision-tree tying. Excluding the "sa" utterances in TIMIT, we use a total of 3696 training utterances and 192 core-test utterances according to the standard setup described in [7].

In the phonetic recognizer's evaluation, we merge the 48 monophones into 39 monophones according to the standard mapping described in [6, 7] and the confusion among the merged phones is not considered as errors.

3.2. P-MCE and Other MCE Methods

Table 1 shows the comparison between the conventional string-based MCE, the lattice-based MCE [2][9] on phone lattices and P-MCE, both using the EBW optimization method of [3]. The number of training iterations is fixed to be five.

The results in Table 1 show that while the string-based MCE reduces phone recognition errors in the training set (compared with the baseline system), it does not achieve the same for the test set. In contrast, the P-MCE technique outperformed the other two MCE methods in terms of phone recognition accuracy on the test set, although for the training set the improvement is not as much as the string-based MCE.

Table 1: *Phone recognition accuracy (percent %) for the conventional string-based MCE, lattice-based MCE and P-MCE*

	Train	Test
Baseline	87.60	72.54
String-based MCE	90.03	70.82
Lattice-based MCE	89.90	72.80
P-MCE	89.64	73.01

3.3. MMI and MPE

The MMI and MPE methods are implemented by the newest HTK3.4 toolkit. The I-smoothed MMI configuration is used for the MMI training, which sets the parameter ISMOOTHTAU=100. The recommended "approximate-error" MPE training (MPE=TRUE, CALCASERROR=TRUE, INSCORRECTNESS=-0.9) is used for the MPE implementation. To make sure MMI and MPE work correctly, we followed the exact procedures on the HTK tutorial and recorded the values of their objective functions over each training iterations.

These values are shown in Table 2 as a function of the training iteration. Consistent increases of the objective functions suggest that the parameters of the algorithms are unlikely to be incorrectly set.

Table 2: *Objective functions of MMI and MPE*

	iter 1	iter2	iter 3	iter 4	iter 5
MMI	0.815	0.923	0.930	0.949	0.955
MPE	0.882	0.916	0.918	0.930	0.941

Table 3 shows the phonetic accuracy results on the core test set after five iterations of MMI and MPE training. The performance of the P-MCE method on the test set is better than the performance of the MMI method but slightly worse than the one of the MPE method. We examined the earlier work on MMI training [4] for the same TIMIT phonetic recognition task. With a much lower baseline performance of phone accuracy of 66.07%, MMI training only improved the accuracy to 67.50%. Based on the trend of the result figures in [4] and extrapolating them to our comparable higher baseline accuracy of 72.54%, very limited improvement from MMI training would be obtained. So our results reported in Table 3 appear to be consistent with those in [4].

Table 3: *Phone recognition accuracy for the conventional string-based MMI and MPE*

	Train	Test
Baseline	87.60	72.54
MMI	89.48	72.85
MPE	89.12	73.03
P-MCE	89.64	73.01

4. Conclusions and future work

In this paper, we introduce a novel MCE training method, P-MCE, which aims at the phone-level discriminative training based on the MCE criterion defined specifically for minimizing phone recognition errors. P-MCE provides a new scheme for selecting competing tokens for each mis-recognized phone in the training utterances, and it takes advantage of the merits associated with both N-best lists and phone lattices. A modified EBW optimization method is described for optimizing the P-MCE objective function.

The conventional wisdom suggests that the MPE criterion is superior to the string-based MCE method as it localizes the training errors more accurately. The P-MCE method allows only one error each utterance, which in fact converts the traditional string-based MCE to a phone-based MCE. Though in our experiments P-MCE shows better performance than MMI and two other MCE methods, while being about the same as the MPE method, the improvement over the (high quality) baseline is lower than expected. One possible cause is the use of 48 phone classes in the training while after folding them into 39 new phone classes, the confusion among the merged phones is not counted as errors [7]. This wastes much of the power in discriminative training, particularly among the phones that are eventually merged into the same class. To correct this problem, a new evaluation standard without phone folding needs to be established.

Our future work will involve extending the current one-best selection of competing tokens (from the lattice) to the richer N-best one within the same P-MCE framework described in this paper. Another obvious improvement of P-MCE will involve training of selective substrings instead of uniformly single phones as reported in this paper. Finally, we plan to incorporate discriminative margins into P-MCE training. This incorporation can be more easily formulated in the P-MCE framework than in other frameworks such as MPE and MMI.

5. Acknowledgements

The authors would like to thank Dr. Hui Jiang and Dr. Dong Yu for useful discussions.

6. References

[1] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 257–265, May 1997.

[2] R. Schluter, W. Macherey, B. Muller, and H. Ney, "Comparison of discriminative training criteria and optimization methids for speech recognition," *Speech Communication*, vol. 34, no. 1, pp. 287–310, May. 2001.

[3] X. He, L. Deng, and W. Chou, "A novel learning method for hidden markov models in speech and audio processing," in *IEEE international workshop on Multimedia Signal Processing (MMSP)*, Victoria, BC, Canada, Oct. 2006.

[4] S. Kapadia, V. Valtchev, and S. J. Young, "Mmi training for continuous phoneme recognition on the timit database," in *ICASSP-93*, Minneapolis, MN, Apr. 1993.

[5] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved dsicriminative training," in *ICASSP-02*, Orlando, FL, May 2002, pp. 105–108.

[6] K.F. Lee and H.W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.

[7] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, vol. 17, no. 2-3, pp. 137–152, 2003.

[8] E. McDermott, *Discriminative training for speech recognition*, Ph.D. thesis, Waseda University, Tokyo, Japan, March 1997.

[9] Q. Fu, A.Moreno, B.-H. Juang, J.-L Zhou, and F.Soong, "Generalization of the minimum classification error (mce) training based on maximizing generalized posterior probability (gpp)," in *ICSLP-2006*, Pittsburgh, PA, Sep. 2006.

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

[11] E. McDermott and S. Katagiri, "Minimum classification error for large scale speech recognition tasks using weighted finite state transducers," in *ICASSP-05*, Philadelphia, PA, March 2005, pp. 113–116.