Direct Error Rate Minimization of Hidden Markov Models

Joseph Keshet[1]*, Chih-Chieh Cheng*[2]*, Mark Stoehr*[3]*, David McAllester*[1]

[1]TTI-Chicago
[2]Department of Computer Science and Engineering, University of California, San Diego
[3]Department of Computer Science, University of Chicago
jkeshet@ttic.edu

Abstract

We explore discriminative training of HMM parameters that directly minimizes the expected error rate. In discriminative training one is interested in training a system to minimize a desired error function, like word error rate, phone error rate, or frame error rate. We review a recent method (McAllester, Hazan and Keshet, 2010), which introduces an analytic expression for the gradient of the expected error-rate. The analytic expression leads to a perceptron-like update rule, which is adapted here for training of HMMs in an online fashion. While the proposed method can work with any type of the error function used in speech recognition, we evaluated it on phoneme recognition of TIMIT, when the desired error function used for training was frame error rate. Except for the case of GMM with a single mixture per state, the proposed update rule provides lower error rates, both in terms of frame error rate and phone error rate, than other approaches, including MCE and large margin.

Index Terms: hidden Markov models, online learning, direct error minimization, discriminative training, automatic speech recognition, minimum phone error, minimum frame error

1. Introduction

Most up-to-date speech recognition systems are based on hidden Markov models (HMMs). The most straightforward technique to estimate the HMM parameters from a training set of examples is using maximum likelihood (ML) estimation. ML estimation attempts to maximize the likelihood of the joint distribution of the observations and the states. It is very efficient and widely used, but it does not directly optimize the performance of these models. Over the years many researchers in speech recognition have studied alternative approaches for parameter estimation, mainly focusing on discriminative training.

Several discriminative methods have been proposed trying to minimize the error rate, including maximum mutual information (MMI) [1], minimum classification error (MCE) [2], minimum word error (MWE), and minimum phone error (MPE) [3]. While all of them lead to lower error rates when properly trained [4, 5, 6], the proposed objective functions all use surrogate smoothed approximations to their respective desired error functions.

Recently, several researchers have proposed methods for large margin training of continuous density hidden Markov models (CD-HMMs) [7, 8, 9]. In large margin training, acoustic models are estimated to assign significantly higher scores to correct transcriptions than competing ones; in particular, the margin between these scores may be required to grow in proportion to the error function. Large margin training achieves good performance, but it does not minimize the error-rate directly.

Recently McAllester, Hazan and Keshet [10] propose to minimize the expected error-rate directly by computing its gradient. This work presented an analytic expression for the gradient of the expected error-rate, which can be used in an update rule for training linear structured prediction models. Here we adapt this method for CD-HMM, which are not linear in the set of parameters. We use this update rule in online training of HMM parameters, similar to large margin update of CD-HMM proposed in [9]. The proposed method can work with any type of error function typically used in speech recognition, such as word error rate, phone error rate and frame error rate. This method was previously tested on phoneme alignment task using the TIMIT dataset. The performance attained surpassed all previously reported results on this problem [10]. A different approach to minimized the regularized error-rate was introduced in [11].

The paper is organized as follows. In Sec. 2, we present the problem setting and the goal. In Sec. 3 we describe two closely-related training approaches, namely, Perceptron and large margin. Our approach is presented in Sec. 4. In Sec. 5 we present results on the TIMIT speech database. Finally, in Sec. 6, we present our conclusions and ideas for future work.

2. Problem Setting

In this section we present the problem definition and our goal. We begin by reviewing the basic notation of a CD-HMM, following the notation presented in [9]. CD-HMMs define a joint probability distribution over a sequence of observations $\mathbf{x} = (x_1, \ldots, x_T)$ and a sequence of (hidden) phonetic states $\mathbf{s} = (s_1, \ldots, s_T)$. The joint distribution is expressed in terms of an initial state distribution $\mathcal{P}(s_1)$, the state transition matrix $\mathcal{P}(s_{t+1}|s_t)$, and the emission densities $\mathcal{P}(x_t|s_t)$. The joint distribution is given by

$$\mathcal{P}(\mathbf{x}, \mathbf{s}) = \mathcal{P}(s_1) \prod_{t=1}^{T-1} \mathcal{P}(s_{t+1}|s_t) \prod_{t=1}^{T} \mathcal{P}(x_t|s_t). \quad (1)$$

The emission densities are assumed to be modeled as Gaussian mixture models (GMMs). Let M denote the number of mixture components in each state. Each mixture m of the s-th state is represented by a multivariate Gaussian with a mean μ_{sm} and a covariance matrix Σ_{sm}. Thus, the emission density function of an observation x given a state s is given by

$$\mathcal{P}(x|s) = \sum_{m=1}^{M} \mathcal{P}(m|s) \mathcal{N}(x; \mu_{sm}, \Sigma_{sm}), \quad (2)$$

where $\mathcal{N}(x; \mu, \Sigma)$ denotes the multivariate Gaussian function at point x and $\mathcal{P}(m|s)$ denotes the mixture weights.

Let us denote by Θ the set of the parameters of the CD-HMM, and let us use the notation $\mathcal{P}(\mathbf{x}, \mathbf{s}|\Theta)$ to emphasize the dependency of the joint distribution on the parameters. The model parameters Θ are estimated from a training set of examples. Each example is composed of a sequence of observations $\mathbf{x}$ and a sequence of target (ground truth) states $\mathbf{y}$. The most likely sequence of states given the sequence of observations $\mathbf{x}$ predicted by the model with parameters Θ is as follows

$$\hat{\mathbf{s}} = \hat{\mathbf{s}}(\mathbf{x}, \Theta) = \arg\max_{\mathbf{s}} \log \mathcal{P}(\mathbf{x}, \mathbf{s}|\Theta), \tag{3}$$

which can be computed efficiently using the Viterbi algorithm. We denote by $\mathcal{L}(\hat{\mathbf{s}}, \mathbf{y})$ the error rate, or the loss, of predicting the state sequence $\hat{\mathbf{s}}$ where the target state sequence is $\mathbf{y}$. Usually, the performance of speech recognition systems is measured in terms of *word error rate*. In this paper, we exemplify our ideas with the problem of phoneme recognition, where the performance is measured in terms of *phone error rate*, that is - the number of substitutions, insertions and deletions, normalized by the number of phones in the target sequence. Sometimes the error rate of phoneme recognition systems is given by *frame error rate*, which is the relative number of frames (states) that are misclassified.

Our goal is to train the model and find its parameters so as to minimize the expected error rate. More formally,

$$\Theta^* = \arg\min_{\Theta} \mathbb{E}\left[\mathcal{L}(\hat{\mathbf{s}}(\mathbf{x}, \Theta), \mathbf{y})\right], \tag{4}$$

when the expectation is over a draw of $(\mathbf{x}, \mathbf{y})$ from a fixed, but unknown distribution. The paper is focused on a new technique to train the model by minimizing the expected error-rate.

To minimize the expected error-rate we aim to perform gradient descent directly on the objective in Eq. (4). Unfortunately, direct gradient descent on Eq. (4) is conceptually puzzling since the state sequence space is discrete. In this case the predicted state sequence $\hat{\mathbf{s}}(\mathbf{x}, \Theta)$ is not a differentiable function of Θ. As one smoothly changes Θ in Eq. (3) the the sequence $\hat{\mathbf{s}}$ jumps discontinuously between discrete state values. So one cannot write $\nabla_\Theta \mathbb{E}[\mathcal{L}(\hat{\mathbf{s}}, \mathbf{y})]$ as $\mathbb{E}[\nabla_\Theta \mathcal{L}(\hat{\mathbf{s}}, \mathbf{y})]$. The main result of this paper is applying our previous work [10] for training CD-HMMs and performing direct gradient descent on Eq. (4).

3. Perceptron-like Training Methods

Our approach builds on earlier work on Perceptron training of discrete HMMs [12] and the Perceptron and large-margin training of continuous density HMMs [13, 9]. We start this section by briefly reviewing the latter before presenting the direct error-rate minimization update rule. Let us define a discriminant function over the observation and state sequences, by taking the logarithm of the joint distribution $\mathcal{P}(\mathbf{x}, \mathbf{s})$ defined in Eq. (1),

$$\mathcal{D}(\mathbf{x}, \mathbf{s}) = \log \mathcal{P}(s_1) + \sum_{t=1}^{T-1} \log \mathcal{P}(s_{t+1}|s_t) + \sum_{t=1}^{T} \log \mathcal{P}(x_t|s_t). \tag{5}$$

Let $\mathbf{y}$ denote the correct transcription of the observation sequence $\mathbf{x}$. Recall that the predicted state sequence $\hat{\mathbf{s}}_n$ is given in Eq. (3). The Perceptron update rule of the CD-HMM parameters is given as

$$\Theta \leftarrow \Theta + \eta \frac{\partial}{\partial \Theta} \left[\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \hat{\mathbf{s}}_n)\right] \tag{6}$$

where $\eta > 0$ is a learning rate. The update in Eq. (6) attempts to close the gap between $\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n)$ and $\mathcal{D}(\mathbf{x}_n, \hat{\mathbf{s}}_n)$ whenever a recognition error occurs.

Large margin training, on the other hand, seeks to separate the scores of correct and incorrect transcriptions by the margin given from the error rate between them. That is

$$\forall \mathbf{s} \neq \mathbf{y}, \quad \mathcal{D}(\mathbf{x}, \mathbf{y}) > \mathcal{D}(\mathbf{x}, \mathbf{s}) + \rho \mathcal{L}(\mathbf{s}, \mathbf{y}); \tag{7}$$

where $\mathcal{L}(\mathbf{s}, \mathbf{y})$ is the error, or loss, incurred between the hidden state sequence $\mathbf{s}$ and the target state sequence $\mathbf{y}$, and $\rho > 0$ is a constant margin scaling factor. In other words, for large margin training, the score of the correct transcription should exceed the score of any incorrect transcription by an amount that grows in proportion to the number of recognition errors. Define the *error-adjusted inference*, $\hat{\mathbf{s}}^\rho$, as follows

$$\hat{\mathbf{s}}^\rho = \arg\max_{\mathbf{s}} \left[\mathcal{D}(\mathbf{x}, \mathbf{s}) + \rho \mathcal{L}(\mathbf{s}, \mathbf{y})\right], \tag{8}$$

The right hand side of Eq. (8) can be maximized by a simple variant of the standard Viterbi algorithm. For online training of large margin CD-HMMs, the following update rule was proposed [13, 9]:

$$\Theta \leftarrow \Theta + \eta \frac{\partial}{\partial \Theta} \left[\mathcal{D}(\mathbf{x}_n, \mathbf{y}_n) - \mathcal{D}(\mathbf{x}_n, \hat{\mathbf{s}}_n^\rho)\right] \tag{9}$$

Eq. (9) differs from Eq. (6) in one critical aspect: namely, we replace the predicted state sequence $\hat{\mathbf{s}}_n$ with the sequence $\hat{\mathbf{s}}_n^\rho$ from margin-based decoding.

4. Direct Error Rate Minimization

Here we consider the following update rule, where ϵ is the error-adjustment weight.

$$\Theta \leftarrow \Theta + \eta \frac{\partial}{\partial \Theta} \left[\mathcal{D}(\mathbf{x}_n, \hat{\mathbf{s}}_n) - \mathcal{D}(\mathbf{x}_n, \hat{\mathbf{s}}_n^\epsilon)\right] \tag{10}$$

where $\hat{\mathbf{s}}^\epsilon$ is defined similarly to $\hat{\mathbf{s}}^\rho$

$$\hat{\mathbf{s}}^\epsilon = \arg\max_{\mathbf{s}} \left[\mathcal{D}(\mathbf{x}, \mathbf{s}) + \epsilon \mathcal{L}(\mathbf{s}, \mathbf{y})\right], \tag{11}$$

whereas the role of ϵ is very different than ρ. While ρ is a margin scaling factor, ϵ is a small number approaching to zero. Note that when $\epsilon = 0$ the update rule is meaningless since both $\hat{\mathbf{s}}^\epsilon$ and $\hat{\mathbf{s}}$ are the same. When we gradually increase ϵ, there is a point where $\hat{\mathbf{s}}_n^\epsilon$ is not equal to $\hat{\mathbf{s}}_n$ – this is exactly the point where the error-rate starts to influence. As we shall see, the update rule is closely related to the definition of the gradient of the expected error-rate. In the update in Eq. (11) we view $\hat{\mathbf{s}}^\epsilon$ as worse than $\hat{\mathbf{s}}$. The update direction moves away from observations adjusted toward the error-rate. Comparing the large-margin update to this update, the target state sequence $\mathbf{y}_n$ in the former is replaced by the inferred state sequence $\hat{\mathbf{s}}$ in the latter.

We now show that under mild conditions the expected update direction of Eq. (10) approaches the negative direction of the gradient of the expected error-rate $\nabla_\Theta \mathbb{E}[\mathcal{L}(\hat{\mathbf{s}}, \mathbf{y})]$ in the limit as the update weight ϵ goes to zero.

Theorem 1 *For a finite set of possible state sequence values, and under reasonable boundary constraints [10], we have the following where $\hat{\mathbf{s}}^\epsilon$ is a function of Θ, $\mathbf{x}$, $\mathbf{y}$ and ϵ.*

$$\nabla_\Theta \mathbb{E}\left[\mathcal{L}(\mathbf{s}, \mathbf{y})\right] = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \mathbb{E}\left[\frac{\partial}{\partial \Theta} \left[\mathcal{D}(\mathbf{x}, \hat{\mathbf{s}}^\epsilon) - \mathcal{D}(\mathbf{x}, \hat{\mathbf{s}})\right]\right] \tag{12}$$

where

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \mathcal{D}(\mathbf{x}, \mathbf{s})$$

and

$$\hat{\mathbf{s}}^\epsilon = \arg\max_{\mathbf{s}} \mathcal{D}(\mathbf{x}, \mathbf{s}) + \epsilon \mathcal{L}(\mathbf{y}, \mathbf{s}).$$

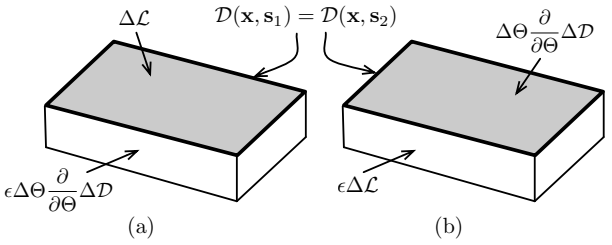

Figure 1: The figure presents the integration involve in computing the expectation of (a) the left hand side and (b) the right hand side of Eq. (12).

A detailed proof of the theorem for the binary linear case is given in [10]. For completeness we give an intuitive sketch of the proof here.

Proof sketch. We start by looking at the left hand side of Eq. (12). Let $u(\epsilon) \sim v(\epsilon)$ be an asymptotic notation for $\lim_{\epsilon\to 0} u/v = 1$. By definition of the gradient we have the following.

$$\epsilon\Delta\Theta \cdot \nabla_\Theta \mathbb{E}\left[\mathcal{L}(\hat{\mathbf{s}}(\mathbf{x},\Theta),\mathbf{y})\right] \sim \mathbb{E}\left[\mathcal{L}(\hat{\mathbf{s}}(\mathbf{x},\Theta+\epsilon\Delta\Theta),\mathbf{y}) - \mathcal{L}(\hat{\mathbf{s}}(\mathbf{x},\Theta),\mathbf{y})\right] \quad (13)$$

Note that $\mathcal{L}(\hat{\mathbf{s}}(\mathbf{x},\Theta+\epsilon\Delta\Theta),\mathbf{y}) - \mathcal{L}(\hat{\mathbf{s}}(\mathbf{x},\Theta),\mathbf{y})$ is nonzero only when $\mathbf{s}_1 = \hat{\mathbf{s}}(\mathbf{x},\Theta+\epsilon\Delta\Theta) \neq \hat{\mathbf{s}}(\mathbf{x},\Theta) = \mathbf{s}_2$. For very small ϵ this happens only when $\mathbf{x}$ is very near the decision boundary between $\mathbf{s}_1$ and $\mathbf{s}_2$ — we have that $\mathbf{x}$ is on the decision boundary between $\mathbf{s}_1$ and $\mathbf{s}_2$ when $\mathcal{D}(\mathbf{x},\mathbf{s}_1) = \mathcal{D}(\mathbf{x},\mathbf{s}_2)$ and both $\mathbf{s}_1$ and $\mathbf{s}_2$ yield the maximum value of $\mathcal{D}(\mathbf{x},\mathbf{s})$. The value of Eq. (13) is determined by an integral over the decision boundaries. Specifically, we integrate the quantity $\Delta\mathcal{L} = \mathcal{L}(\mathbf{s}_1,\mathbf{y}) - \mathcal{L}(\mathbf{s}_2,\mathbf{y})$ times the width of the region where the state sequence switches when Θ is changed by $\epsilon\Delta\Theta$. The width of the region where the state sequence switches is $\epsilon\Delta\Theta \cdot \frac{\partial}{\partial\Theta}\Delta\mathcal{D}$ where $\Delta\mathcal{D} = \mathcal{D}(\mathbf{x},\mathbf{s}_1) - \mathcal{D}(\mathbf{x},\mathbf{s}_2)$. This integral is show pictorially in the left hand side of Fig. 1.

To show Eq. (12) it suffices to show that the quantity in Eq. (13) is asymptotically equivalent to the following in the limit of small ϵ.

$$\Delta\Theta \cdot \mathbb{E}\left[\frac{\partial}{\partial\Theta}\Big(\mathcal{D}(\mathbf{x},\hat{\mathbf{s}}^\epsilon(\mathbf{x},\Theta)) - \mathcal{D}(\mathbf{x},\hat{\mathbf{s}}(\mathbf{x},\Theta))\Big)\right]$$

This quantity is nonzero only when $\mathbf{s}_1 = \hat{\mathbf{s}}^\epsilon(\mathbf{x},\Theta) \neq \hat{\mathbf{s}}(\mathbf{x},\Theta) = \mathbf{s}_2$. As ϵ goes to zero $\hat{\mathbf{s}}^\epsilon(\mathbf{x},\Theta) \neq \hat{\mathbf{s}}(\mathbf{x},\Theta)$ only when $\mathbf{x}$ is very near the decision boundary between $\mathbf{s}_1$ and $\mathbf{s}_2$. As in the previous case, the value of right hand side of Eq. (12) is determined by an integral over decision boundaries. In this case we case we integrate the quantity $\Delta\Theta \cdot \frac{\partial}{\partial\Theta}\Delta\mathcal{D}$ times the width of the region where the error adjustment switches the state sequence. The width of the region where the error adjustment switches the state sequence is $\epsilon\Delta\mathcal{L}$. This is show pictorially in the right hand side of Fig. 1. The integrals shown in the left and right hand side of Fig. 1 are both equal to the integral over the decision boundary of $\epsilon\Delta\mathcal{L}\Delta\Theta \cdot \frac{\partial}{\partial\Theta}\Delta\mathcal{D}$. A more complete proof can be found in [10].

5. Experiments

We evaluated the proposed method on the TIMIT acoustic-phonetic continuous speech corpus [14]. The training set contains 462 speakers and 3696 utterances. We used the core test set of 24 speakers and 192 utterances and a development set of 50 speakers and 400 utterances as defined in [7] for tuning the parameters. Following the common practice [15], we mapped the 61 TIMIT phonemes into 48 phonemes for training, and further collapsed from 48 phonemes to 39 phonemes for evaluation. We extracted the standard 12 MFCC features and log energy with their deltas and double deltas to form 39-dimensional acoustic feature vectors. The window size and the frame size were 25 msec and 10 msec, respectively.

We built recognizers using monophone CD-HMMs in which each of 48 states represented a context-independent phoneme, and we used the GMM re-parametrization as described in [8]. We experimented with models of different sizes by varying the number of Gaussian mixture components in each state. We evaluated the performance of each CD-HMM by comparing the hidden state sequences inferred by Viterbi decoding to the "ground-truth" phonetic transcriptions provided by the TIMIT corpus. We report two types of errors: the frame error rate (FER), computed simply as the percentage of misclassified frames, and the phone error rate (PER), computed from the edit distances between ground truth and Viterbi decodings. In calculating the errors, we follow the standard of mapping 48 phonetic classes down to broader 39 categories [15]. The performance of our baseline models with maximum likelihood estimation is similar to those previously reported [7, 9].

All CD-HMMs were initialized by maximum likelihood estimation. Starting from these baseline CD-HMMs, we then compared the performance of the different online updates in Eq. (9) and Eq. (10). For the margin-based update (9), the results of training depend on the margin scaling factor ρ. We chose the scaling factor ρ that yielded the lowest phone error rates on the held-out development set.

For the direct error-rate minimization update (10), we used the error function $\mathcal{L}$ to be *frame error rate* as it is easier to implement compared to *phone error rate*. Our method is not limited to this type of error function and we defer the use of phone error rate to future work. The training with the direct error-rate minimization update rule depends on ϵ. While ϵ theoretically should approach zero, the way we perform the optimization was by picking the ϵ that yielded the lowest phone error rates on the held-out development set ($\epsilon = 0.65$). At each iteration we computed the potential update for the given utterance. The update was only adopted if it decreased the error for that specific example. Since the direct error-rate minimization is based on two predictions, $\hat{\mathbf{s}}$ and $\hat{\mathbf{s}}^\epsilon$, it is likely to get stuck in a local minimum. In order to avoid local minima, when applying the direct error-rate minimization update rule (10), we iterated it with a large-margin update rule (9), which is the closest update rule, in terms of the objective. For a fair comparison we used the same number of iterations in all of the online methods.

In general, this mistake-driven approach will not converge to a fixed set of parameters. However, convergence to a fixed set can be obtained by averaging parameters across different updates of Eq. (9) and Eq. (10); the averaging also gives a better result after a finite number of iterations through the training set [16]. Note that this averaging does not affect training process: it only affects the parameters used for evaluating the model on held-out data.

We present the results in terms of frame error rate of maximum likelihood (ML) estimation, online and batch large margin (LM) training and direct error-rate minimization in Table 1. The results of ML, online LM and batch LM are reproduced from previous benchmarks [17, 8]. The results show that direct error-rate minimization reduces the frame error rates across all

Table 1: Frame error rates on the TIMIT test set as obtained by maximum likelihood (ML) estimation, online and batch of large margin training, and direct error-rate minimization. The results in the first three columns are reproduced from previous benchmarks [17, 8].

# mixtures	Frame Error Rate (%)			
	ML	online LM	batch LM	**direct error-rate min.**
1	39.7	30.5	**29.5**	30.1
2	36.2	29.4	29.0	**27.8**
4	33.1	28.3	28.4	**27.0**
8	30.7	27.3	27.2	**26.4**
16	29.5	27.3	-	**26.3**
32	29.9	27.6	-	**26.7**

Table 2: Phone error rates on the TIMIT test set as obtained by maximum likelihood (ML) estimation, minimum classification error (MCE), online and batch of large margin training, and direct error-rate minimization. The results in the first four columns are reproduced from previous benchmarks [17, 8].

# mixtures	Phone Error Rate (%)				
	ML	MCE	online LM	batch LM	**direct error-rate min.**
1	41.5	35.6	32.8	**31.2**	32.5
2	38.0	34.5	31.4	30.8	**30.1**
4	34.9	32.4	30.3	29.8	**29.4**
8	32.3	30.9	28.6	28.2	**27.6**
16	30.8	-	28.8	-	**28.1**
32	31.8	-	29.0	-	**28.3**

model sizes except for a single mixture, where the large margin batch is better. This suggest that the optimization for the first mixture didn't reach the optimum value.

We present result in terms of phone error rate of maximum likelihood (ML) estimation, minimum classification error (MCE), online and batch large margin (LM) training and direct error-rate minimization in Table 2. The results of ML, MCE, online LM and batch LM are all reproduced from previous benchmarks [17, 8]. Again the results suggest that direct error-rate minimization reduces the phone error rates across all model sizes except for a single mixture, where the large margin batch is better. These results for are also comparable or better than previously published benchmarks for batch implementations of discriminative training on this task [8]. In general, the frame error rates improve more than the phone error rates; this discrepancy reflects the fact that we used the frame error rate as a our error function.

6. Discussion and Future Work

In this paper we proposed a new type of online update to train CD-HMMs. The update directly minimizes a user-defined error function, which can be non-convex and non-differentiable, as long as the expected error function is smooth and differentiable in the set of parameters, such as frame error rate, phone error rate, and word error rate. We presented a theorem with a proof sketch, which states that the gradient of the expected error-function equals to the derivative of the difference between the two functions of the log of joint probability: one with the inferred state sequence and the other with the error-adjusted inferred state sequence. We presented experiments when the frame error-rate used as the objective error function. The results suggest that the direct error-rate minimization outperformed all types of training criteria, including batch LM, except for GMM with a single mixture per state.

Future work will be focused on using phone error rate as the objective error function. In the presented experiments we did not compare our method to MPE. We also defer this important comparison to future work. Lastly, we would like to check the efficiency of the proposed training on a large scale tasks.

Acknowledgement. Chih-Chieh Cheng is supported by NSF Award 0812576.

7. References

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

[2] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans on Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.

[3] D. Povey and P. Woodland, "Minimum phone error and I-Smoothing for improved discriminative training," in *Proc. of ICASSP*, 2002.

[4] J. Roux and E. McDermott, "Optimization methods for discriminative training," in *Proc. of Interspeech*, 2005.

[5] E. McDermott, T. Hazen, J. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using minimum classification error," *IEEE Trans on Speech and Audio Processing*, vol. 15, no. 1, pp. 203–223, 2006.

[6] E. McDermott, S. Watanabe, and A. Nakamura, "Discriminative training based on an integrated view of MPE and MMI in margin and error space," in *Proceedings of ICASSP*, 2010.

[7] F. Sha and L. K. Saul, "Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models," in *Proc. of ICASSP*, 2007, pp. 313–316.

[8] ——, "Large margin training of continuous density hidden Markov models," in *Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods*, J. Keshet and S. Bengio, Eds. Wiley and sons, 2009, pp. 101–114.

[9] C.-C. Cheng, F. Sha, and L. K. Saul, "A fast online algorithm for large margin training of continuous-density hidden Markov models," in *Proc. of Interspeech*, 2009.

[10] D. A. McAllester, T. Hazan, and J. Keshet, "Direct loss minimization for structured prediction," in *Proc. of NIPS 24*, 2010.

[11] J. Keshet, D. McAllester, and T. Hazan, "PAC-Bayesian approach for minimization of phoneme error rate," in *Proc. of ICASSP*, 2011.

[12] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *Proc. of EMNLP*, 2002.

[13] C.-C. Cheng, F. Sha, and L. K. Saul, "Matrix updates for Perceptron training of continuous density hidden Markov models," in *Proc. of ICML*, 2009.

[14] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design an analysis of the acoustic-phonetic corpus," in *DARPA Speech Recognition Workshop*, 1986.

[15] K.-F. Lee and H.-W. Hon, "Speaker independent phone recognition using hidden markov models," *IEEE Trans. Acoustic, Speech and Signal Proc.*, vol. 37, no. 2, pp. 1641–1648, 1989.

[16] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *IEEE Trans. on Information Theory*, vol. 50, no. 9, pp. 2050–2057, September 2004.

[17] C.-C. Cheng, F. Sha, and L. K. Saul, "Online learning and acoustic feature adaptation in large margin hidden Markov models," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 6, pp. 926–942, 2010.