Discriminative Training using the Trusted Expectation Maximization

Yasser Hifny, Yuqing Gao

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
{yhifny,yuqing}@us.ibm.com

Abstract
We present the Trusted Expectation-Maximization (TEM), a new discriminative training scheme, for speech recognition applications. In particular, the TEM algorithm may be used for Hidden Markov Models (HMMs) based discriminative training. The TEM algorithm has a form similar to the Expectation-Maximization (EM) algorithm, which is an efficient iterative procedure to perform maximum likelihood in the presence of hidden variables [1]. The TEM algorithm has been empirically shown to increase a rational objective function. In the concave regions of a rational function, it can be shown that the maximization steps of the TEM algorithm and the hypothesized EM algorithm are identical. In the TIMIT phone recognition task, preliminary experimental results show competitive optimization performance over the conventional discriminative training approaches (in terms of speech and accuracy).

Index Terms: Expectation-Maximization (EM), Discriminative training, Speech Recognition

1. Introduction
Acoustic modeling based on Hidden Markov Models (HMMs) are the choice for state-of-the-art stochastic speech recognition applications. HMMs are well understood models and may be trained effectively using the Expectation-Maximization (EM) algorithm [1]. The EM algorithm is widely used for density estimation problems. Discriminative training implies a separation measure, in terms of a rational function, between the correct class and all competing classes. Hence, the EM algorithm cannot be used for discriminative training. In this work, an alternative for the EM algorithm is proposed for rational functions. We refer to it as the Trusted Expectation-Maximization (TEM) scheme.

In Section 2, an overview on HMM generative and discriminative training is presented. HMM discriminative training based on the Trusted Expectation-Maximization (TEM) algorithm is detailed in Section 3. Section 4 describes a convex smoothing approach to improve the generalization of the TEM algorithm. Sections 5 and 6 give experimental results on the TIMIT phone recognition task and conclusions.

2. HMM optimization
HMMs can be trained to maximize the likelihood of the data given the underlying parameterized distributions using the Expectation-Maximization (EM) algorithm [1]. Generative training is widely used in speech recognition because it is fast and efficient. If the true distribution that generated the data is indeed an HMM, then, given sufficient data, Bayes classification based on HMMs estimated using maximum likelihood will minimize the probability of classification error [2]. However, HMM discriminative training may be used to improve accuracy and utilize the parameters efficiently [3, 4, 5, 6, 7, 8].

2.1. Generative parameter estimation
Parameters of HMMs can be estimated using the maximum likelihood (ML) criterion. For R training observations \( \{O_1, O_2, \ldots, O_r, \ldots, O_R\} \) with corresponding transcriptions \( \{w_r\} \), the ML objective function is given by

\[
F_{MLE}(\Lambda) = \sum_{r=1}^{R} \log p(O_r|\mathcal{M}_{w_r}) \tag{1}
\]

where \( \mathcal{M}_{w_r} \) is the composite model corresponding to the reference word sequence \( w_r \).

The parameters can be estimated using the iterative Baum-Welch algorithm, also known as the forward-backward algorithm [9]. The Baum-Welch algorithm is a special case of the Expectation-Maximization (EM) algorithm, which is an efficient iterative procedure to perform ML in the presence of hidden variables [1]. The inference of an HMM is based on computing the forward and backward probabilities. The frame-state alignment probability \( \gamma_{jt} \), denoting the probability of being in state \( j \) at some time \( t \) can be written in terms of the total probability \( p(O|M) \), the forward probability \( \alpha_j(t) \) and the backward probability \( \beta_j(t) \):

\[
\gamma_{jt} = P(s_t = j|O; M) = \frac{p(O, s_t = j|M)}{p(O|M)} = \frac{\alpha_j(t)\beta_j(t)}{p(O|M)} \tag{2}
\]

and a component specific alignment probability can be derived:

\[
\gamma_{jm}(t) = P(s_t = j, m_t = m|O; M) = \gamma_{jt} c_{jm} b_{jm}(o_t) b_j(o_i) \tag{3}
\]

where the \( m \)th component is associated with the \( j \)th state. \( c_{jm} \) is the component weight, \( b_{jm}(o_t) = N(o_t; \mu_{jm}, \Sigma_{jm}) \) is a normal distribution score, and \( b_j(o_i) = \sum_{m=1}^{M} c_{jm} b_{jm}(o_t) \) is an output distribution score. \( \mu_{jm} \) and \( \Sigma_{jm} \) are the component specific mean vector and covariance matrix respectively.

Consequently, the accumulators of the sufficient statistics \( C_{jm}(1), C_{jm}(O), \) and \( C_{jm}(O^2) \) are calculated as follows:

\[
C_{jm}(1) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{jm}^r(t) \tag{4}
\]

\[
C_{jm}(O) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{jm}^r(t) o_t \tag{5}
\]

\[
C_{jm}(O^2) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_{jm}^r(t) o_t^2 \tag{6}
\]
where $C_{jm}(1) = \gamma_{jm}$.

Hence, the EM re-estimation formulae for the mean and covariance of state $j$ and component $m$ of an HMM are given by

$$
\mu_{jm} = \frac{C_{jm}(O)}{C_{jm}(1)}
$$

$$
\Sigma_{jm} = \frac{C_{jm}(O^2)}{C_{jm}(1)} - \mu_{jm} \mu_{jm}^T
$$

The transition probabilities between states are also estimated by calculating the forward and backward probabilities. The full maximization step of the EM algorithm does not have hyperparameters to tune. This attractive property may be the main reason behind the success of HMM generative training.

HMM generative training leads to models that may be useful for generating speech, which is useful for speech synthesis. Using Bayes rule, these generative models can be used for speech recognition.

### 2.2. Discriminative parameter estimation

HMMs trained using the EM algorithm are very effective for coarse generation of data. Unfortunately, generative training does not address the classification problem, where the objective is to discriminate between the classes and hence to reduce the misclassification error.

The Conditional Maximum Likelihood (CML) criterion, defined by equation (9), aims to maximize the log of posterior probability of the correct word sequence corresponding given the observations,

$$
\mathcal{F}_{CML} (\Lambda) = \sum_{r=1}^{R} \log p_{\Lambda}(\mathcal{M}_{w_r} | O_{r})
$$

$$
= \sum_{r=1}^{R} \log \frac{p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}) P(w_r)}{\sum_{w} p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w}) P(w)}
$$

$$
\approx \sum_{r=1}^{R} \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{num}) - \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{den})
$$

where $\mathcal{M}_{w}$ is a composite model corresponding to the word sequence $w$ and $P(w)$ is the probability of this sequence as determined by a language model. This discriminative training aims to maximize the probability of the correct models (known as the numerator) $p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{num})$, which is identical to the ML objective function, and simultaneously minimize the recognition model probability (known as the denominator term) $p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{den}) \approx \sum_{w} p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w}) P(w)$, which is the summation over all possible word sequences $\hat{w}$ allowed in the task, is computationally expensive for most tasks. As a result, $p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{num})$ is an approximation to the denominator term, which is computed by N-Best lists [10] or lattices [11, 12] generated from a decoding pass based on ML trained models.

Extended Baum-Welch (EBW) algorithm is the state-of-the-art discriminative training algorithm that maximize the CML criterion for HMMs. It was introduced for discriminative training for discrete distributions in [5]. Using a discrete approximation to the Gaussian distribution [6], it was shown that the mean of the Gaussian for state $j$, mixture component $m$, $\mu_{jm}$ and the corresponding covariance, $\Sigma_{jm}$ (assuming diagonal covariance matrices) can be reestimated by

$$
\hat{\mu}_{jm} = \frac{\epsilon_{num}^{num}(O) - \epsilon_{den}^{num}(O) + D_{jm} \hat{\mu}_{jm}}{\gamma_{jm} + \gamma_{num}^{num} + D_{jm}}
$$

$$
\hat{\Sigma}_{jm} = \frac{\epsilon_{num}^{num}(O^2) - \epsilon_{den}^{num}(O^2) + D_{jm} (\Sigma_{jm} + \mu_{jm} \mu_{jm}^T) - \hat{\mu}_{jm} \hat{\mu}_{jm}^T}{\gamma_{jm} + \gamma_{num}^{num} + D_{jm}}
$$

where $D_{jm}$ is a smoothing constant that controls the degree of deviation of the new parameters with respect to the old parameters. The superscripts $num$ and $den$ refer to the model corresponding to the correct word sequence, and the recognition model for all word sequences, respectively. Setting the optimal value for $D_{jm}$ is the subject of extensive research and a widely used heuristic is given by

$$
D_{jm} = \max \{ 2D_{jm}^{num}, E_{jm}^{den} \},
$$

where $D_{jm}^{num}$ is a necessary value to ensure positive variances and $E$ is a global constant [13]. Using the reverse Jensen inequality for c-family distributions [14], a closed form expression for $D_{jm}$, was derived and the heuristic in equation (12) was justified [15]. The discriminative training of HMMs is usually initialized by ML generative training. For historical reasons, CML discriminative training for HMMs is known as Maximum Mutual Information Estimation (MMIE) in the speech recognition domain. The two criteria lead to the same results because the language model parameters are not optimized during the training.

### 3. The TEM algorithm

The EM algorithm was derived for the ML criterion and can not be used for the CML discriminative criterion (rational function). The maximization step (M step) may lead to a decrease of the rational objective function value and as well the variances may be negative. To handle this problem, we propose a regularization framework to get meaningful updates. The new algorithm will be very similar to the original EM algorithm. In general, regularization is a common approach to overcome poor generalization and to provide effective complexity control.

#### 3.1. Regularized CML criterion

In this work, regularization is achieved by adding a penalty term to the CML criterion as shown in equation (13).

$$
\mathcal{F}_{RCML} (\Lambda) = \sum_{r=1}^{R} \left( \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{num}) - \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{den}) + \beta \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{den}) \right)
$$

where the penalty term $\beta \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{den})$ has a special form (same as the CML denominator but with a reverse sign). $\beta$ is a hyperparameter which controls the contribution of the penalty term during the optimization process and it is a positive number.

The special penalty term $\beta \log p_{\Lambda}(\mathcal{O}_{r} | \mathcal{M}_{w_r}^{den})$ is introduced in order to derive simple update equations. Due to this special form, the $\beta$ dynamic range is limited (i.e. $0.0 < \beta < 1.0$). This restriction prevents the change of the sign of the overall denominator term. In order to have discriminative training, the overall denominator term must have a negative value.
3.2. RCML optimization

The update equations of the RCML criterion based on equation (13) are very similar to the update equations of the EM algorithm. The re-estimation formulae for the mean and covariance of state \( j \) and component \( m \) of an HMM are given by

\[
\mu_{jm} = \frac{C_{jm}^{\text{num}}(O) - \gamma_{jm}^{\text{den}}(O) + \beta_{jm} \gamma_{jm}^{\text{old}}(O)}{\gamma_{jm}^{\text{num}} - \gamma_{jm}^{\text{den}} + \beta_{jm} \gamma_{jm}^{\text{old}}(O)}
\]

\[
\Sigma_{jm} = \frac{C_{jm}^{\text{num}}(O^2) - \gamma_{jm}^{\text{den}}(O^2) + \beta_{jm} \gamma_{jm}^{\text{old}}(O^2)}{\gamma_{jm}^{\text{num}}(O^2) - \gamma_{jm}^{\text{den}}(O^2) + \beta_{jm} \gamma_{jm}^{\text{old}}(O^2)} - \mu_{jm}\mu_{jm}^T
\]

where \( \beta_{jm} \) is selected at Gaussian level. We refer to these equations as the TEM algorithm update equations. When \( \beta_{jm} = 0 \), the TEM and the hypothesized EM (i.e. the EM algorithm does not provide update equations for rational functions) algorithms have identical re-estimation formulae. However, \( \beta_{jm} \) must be selected to ensure the variances are positive and this is addressed in the next section. The update equations of the TEM algorithm for transition probabilities and mixture weights will not be addressed in this paper.

The regularization framework construct a model trust region around the current model estimate. The amount of regularization is trusted only in a small region around the current estimate (variances must be positive). Hence, the TEM algorithm effectively is an EM algorithm with a model trust region control.

3.3. Critical regularization

The critical regularization is a process which aims to find the hyperparameter \( \beta_{jm} \) values. \( \beta_{jm} \) in equation (11) controls the behavior of the TEM algorithm as well as the contribution of the penalty term. During the optimization, the dynamic range of \( \beta_{jm} \) is limited to \( \beta_c < \beta_{jm} < 1.0 \). The critical regularization parameter, \( \beta_c \), is the minimum value of \( \beta_{jm} \) to ensure the variances of any Gaussian are positive.

Once the sufficient statistics (first and second order moments) are accumulated over the training data, the value of \( \beta_c \) may be computed by solving a quadratic equation for each dimension in each Gaussian. Let \( x = 1 - \beta_c \) and for variances to be positive they must satisfy

\[
ax^2 + bx + c = 0
\]

where

\[
a = \gamma_{jm}^{\text{den}} - \gamma_{jm}^{\text{num}}(O^2) - \gamma_{jm}^{\text{old}}(O^2)
\]

\[
b = 2\gamma_{jm}^{\text{num}}(O)\gamma_{jm}^{\text{den}}(O) - \gamma_{jm}^{\text{num}}(O^2) - \gamma_{jm}^{\text{den}}(O^2) - \gamma_{jm}^{\text{num}}(O)^2
\]

\[
c = \gamma_{jm}^{\text{num}}(O^2) - \gamma_{jm}^{\text{old}}(O)^2
\]

Using simple manipulations, it is possible to get the quadratic equation of \( \beta_c \). In practice, we add a small value \( \delta_{jm} \) defined empirically to compute the actual value of \( \beta_{jm} \) during the optimization. Note that, \( \beta_c + \delta_{jm} \) must be less than 1.0 to have effective discriminative training. Note that, the condition \( \beta_c > 1 - \frac{\gamma_{jm}^{\text{num}}}{\gamma_{jm}^{\text{old}}} > 0.0 \) must be valid during the optimization.

Based on an empirical observation, it was found that \( \beta_c = 0.0 \) for many Gaussians (when \( \gamma_{jm}^{\text{num}} > \gamma_{jm}^{\text{old}} \)). Theoretically, this may mean that the TEM algorithm is identical to the hypothesized EM algorithm in some cases (concave regions). Hence, the TEM algorithm alternates between the hypothesized EM algorithm and a regularized version of it during the optimization process. The alternation process depends on the values of \( \gamma_{jm}^{\text{num}} \) and \( \gamma_{jm}^{\text{old}} \) assigned for each Gaussian. This may indicate that the TEM may be extremely fast algorithm to fit the training data. Although this fact is interesting theoretically, it may harm the generalization (test set performance). Improving the generalization of the TEM algorithm will be the subject of the next section.

4. Convex smoothing

Smoothing is a process that aims to prevent or slow down large changes in the values of HMM parameters during the discriminative optimization. For example, l-smoothing may be used to improve the generalization of discriminative training as described in [16].

In this work, it was found experimentally that the TEM algorithm may fit the training data during the optimization process. Hence, the generalization performance degrades especially after two or three iterations. To handle this problem, we propose to use a convex smoothing method to improve generalization. This method may prevent the TEM algorithm to jump quickly in the parameter space and avoid fitting the training data.

The convex smoothing formulae for the mean and covariance of state \( j \) and component \( m \) of an HMM are given by

\[
\mu_{jm} = (1 - \epsilon)\mu_{jm}^{\text{TEM}} + \epsilon\mu_{jm}^{\text{old}}
\]

\[
\Sigma_{jm} = (1 - \epsilon)\Sigma_{jm}^{\text{TEM}} + \epsilon\Sigma_{jm}^{\text{old}}
\]

where \( 0.0 < \epsilon < 1.0 \) and the interpolation is convex. \( \mu_{jm}^{\text{old}} \) and \( \Sigma_{jm}^{\text{old}} \) are the previous iteration estimate of the parameters and \( \mu_{jm}^{\text{TEM}} \) and \( \Sigma_{jm}^{\text{TEM}} \) are the current iteration estimate of the TEM algorithm.

The \( \epsilon \) should be large (\( \geq 0.5 \)) to slow down the optimization speed. Using this scheme, it was found we slow down the improvements in recognition performance in early iterations but the generalization of the TEM algorithm is improved.

5. Experiments

We have carried out phone recognition experiments on the TIMIT corpus. We used the 462 speaker training set and testing on the 168 speaker full test set (the SA1 and SA2 utterances were not used). The speech was analyzed using a 25ms Hamming window with a 10 ms fixed frame rate. In all the experiments we represented the speech using 12th order mel frequency cepstral coefficients (MFCCs), energy, along with their first and second temporal derivatives, resulting in a 39 element feature vector. Following Lee [17], the original 61 phone classes in TIMIT were mapped to a set of 48 labels, which were used for training. This set of 48 phone classes was mapped down to a set of 39 classes [17], after decoding, and phone recognition results are reported on these classes, in terms of the phone error rate (PER), which is analogous to word error rate. All our experiments used a bigram language model over phones, estimated from the training set. The language model scaling factor is set to 6.0 during the decoding process.

The baseline HMMs have three emitting states and the emission probabilities were modeled with mixtures of Gaussian densities with diagonal covariance matrices. The generative HMMs were trained by the maximum likelihood criterion

1http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1
Table 1: Optimization performance of the TEM and EBW on TIMIT recognition task in terms of PER.

<table>
<thead>
<tr>
<th>Complexity</th>
<th>2 mixture</th>
<th>10 mixture</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML baseline</td>
<td>40.2%</td>
<td>32.6%</td>
</tr>
<tr>
<td>#it</td>
<td>EBW</td>
<td>TEM</td>
</tr>
<tr>
<td>1</td>
<td>38.6%</td>
<td>38.4%</td>
</tr>
<tr>
<td>2</td>
<td>37.7%</td>
<td>37.1%</td>
</tr>
<tr>
<td>3</td>
<td>36.9%</td>
<td>36.3%</td>
</tr>
<tr>
<td>4</td>
<td>36.3%</td>
<td>35.5%</td>
</tr>
<tr>
<td>5</td>
<td>35.9%</td>
<td>35.2%</td>
</tr>
<tr>
<td>6</td>
<td>35.6%</td>
<td>35.0%</td>
</tr>
<tr>
<td>7</td>
<td>35.2%</td>
<td>34.9%</td>
</tr>
<tr>
<td>8</td>
<td>35.0%</td>
<td>34.9%</td>
</tr>
</tbody>
</table>

using the conventional EM algorithm. HMMs were refined using CML discriminative training based on the EBW algorithm [6]. Only the means $\mu_{jm}$ and variances $\sigma_{jm}$ were updated during the CML discriminative training. The EBW learning rate constant $D_{jm} = \max(2F_{jm}^{\min}, \gamma_{jm})$ is used, where $D_{jm}^{\min}$ is the value required to ensure positive variances [8].

In Table 1, we compare the performance of the TEM algorithm with the EBW algorithm for eight iterations. The $\beta$ is computed as described in section 3.3 per each Gaussian. The smoothing hyperparameter $\epsilon$ was to 0.9 in this work and it is necessary to improve the TEM generalization performance. The results show that the TEM algorithm outperforms the EBW in every iteration during the optimization for the two complexity setups. For example, the PER of the TEM algorithm after two iterations of 30.7% is very similar to four iterations performance of the EBW algorithm of 30.6%. Hence, these initial experiments may indicate that the TEM algorithm may provide improvements in accuracy and speed over the conventional EBW discriminative training.

### 6. Conclusion

In this paper we present the TEM algorithm for HMM discriminative training. The update formulae of the new algorithm is very similar to the conventional EM algorithm update. During the optimization process the TEM algorithm alternate between the hypothesized EM algorithm and its regularized version providing an efficient update. Preliminary results show that the TEM algorithm may be competitive with the traditional discriminative training based on the EBW algorithm. Our future work will concentrate on large scale discriminative training and the optimal tuning of the hyperparameters associated with the algorithm.

### 7. References


