Generalized Baum-Welch Algorithm and Its Implication to a New Extended Baum-Welch Algorithm

Roger Hsiao and Tanja Schultz

InterACT, Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{wrhsiao, tanja}@cs.cmu.edu

Abstract

This paper describes how we can use the generalized Baum-Welch (GBW) algorithm to develop better extended Baum-Welch (EBW) algorithms. Based on GBW, we show that the backoff term in the EBW algorithm comes from KL-divergence which is used as a regularization function. This finding allows us to develop a fast EBW algorithm, which can reduce the time of model space discriminative training by half, without incurring any degradation on recognition accuracy. We compare the performance of the new EBW algorithm with the original one on various large scale systems including Farsi, Iraqi and modern standard Arabic ASR systems.

Index Terms: speech recognition, discriminative training

1. Introduction

Model estimation in speech recognition is often formulated as an optimization problem. Common optimization algorithms include the Baum-Welch (BW) algorithm and the extended Baum-Welch (EBW) algorithm [1]. The BW algorithm maximizes the likelihood of the hidden Markov model (HMM) on the train data, while the EBW algorithm optimizes HMM for some discriminative objective functions such as boosted maximum mutual information (BMMI) [2]. Compared to the BW algorithm, the EBW algorithm is more expensive since the discriminative objective functions involve not only the references of the data, but also the competitors. Although the computational cost is much higher, the EBW algorithm with a proper discriminative objective function often outperforms systems trained with maximum likelihood (ML) estimation [2].

In [3], we proposed the generalized Baum-Welch (GBW) algorithm, which is a generalization of the BW and the EBW algorithms. We found that the backoff term (we called it the D-term) in the EBW update equations comes from a distance based regularization in the optimization problem. This is not obvious in the original derivation of the EBW algorithm, and the GBW algorithm can also explain the heuristics used in the EBW algorithm.

The purpose of this paper is to show that the GBW algorithm provides a platform to develop better EBW algorithms. In this paper, we extend [3] and show that the regularization in the original EBW algorithm is based on KL-divergence. Given this piece of information, we demonstrate how to develop a fast EBW algorithm which can achieve the same recognition accuracy with only half the training time. We would like to emphasize that while this new EBW algorithm is useful, this is only one example about how we can use the GBW algorithm to improve the EBW algorithm.

This paper is organized as follows: in section 2, we review the GBW algorithm and regularization. In section 3, we show that the regularization is based on KL-divergence and cross entropy, and explain how we can improve the EBW algorithm. In section 4, we report experimental results on the EBW and our proposed EBW algorithm. We conclude our work and discuss future work in section 5.

2. Generalized Baum-Welch Algorithm

Instead of directly optimizing a discriminative objective function, GBW minimizes,

$$G(X,\theta) = \sum_i |Q_i(X,\theta) - C_i| + R(\theta,\theta^0)\ . \tag{1}$$

where i is an index referring to the reference or the competitor of some utterance; X is the observation; θ represents the model parameters; Q_i is an auxiliary function representing the negative log likelihood and C_i is the target value that we want Q_i to achieve. By setting C_i appropriately, minimizing G is equivalent to optimizing the discriminative objective function. For instance, if the objective function is mutual information, one can set C_i such that $Q_i > C_i$ for all i correspond to references and $Q_i < C_i$ for all i correspond to competitors (i.e. lattices). $R(\theta,\theta^0)$ is an optional regularization function with θ^0 as a backoff model.

Suppose we optimize the mean vectors and we use Mahalanobis distance for regularization, the problem becomes,

$$\begin{aligned} \min_{\epsilon,\mu} \quad & \textstyle\sum_i \epsilon_i & & + \sum_j \frac{D_j}{2}||\mu_j - \mu_j^0||^2_{\Sigma_j} \\ \text{s.t.} \quad & \epsilon_i \geq Q_i(\mu) - C_i & \forall i & \\ & \epsilon_i \geq C_i - Q_i(\mu) & \forall i\ , & \end{aligned} \tag{2}$$

where ϵ_i is a slack variable; D_j is a Gaussian specific constant to control the weight of regularization and μ_j^0 is the backoff mean vector. Then, we construct the Lagrangian,

$$\begin{aligned} L_m(\epsilon,\mu,\alpha,\beta) \quad = \quad & \sum_i \epsilon_i - \sum_i \alpha_i(\epsilon_i - Q_i(\mu) + C_i) \\ - \quad & \sum_i \beta_i(\epsilon_i - C_i + Q_i(\mu)) \\ + \quad & \sum_j \frac{D_j}{2}||\mu_j - \mu_j^0||^2_{\Sigma_j} \end{aligned} \tag{3}$$

where $\{\alpha_i\}$ and $\{\beta_i\}$ are the Lagrange multipliers for the first and the second set of constraints of the optimization problem in

equation 2. The Lagrangian dual is then defined as,

$$L_m^D(\alpha,\beta) = \inf_{\epsilon,\mu} L_m(\epsilon,\mu,\alpha,\beta)\ . \tag{4}$$

Now, we can differentiate L_m w.r.t. μ and ϵ. Hence,

$$\begin{aligned}\frac{\partial L_m}{\partial \epsilon_i} &= 1-\alpha_i-\beta_i &(5)\\ \frac{\partial L_m}{\partial \mu_j} &= \sum_i(\alpha_i-\beta_i)\frac{\partial Q_i}{\partial \mu_j} + D_j\frac{\partial}{\partial \mu_j}||\mu_j-\mu_j^0||^2_{\Sigma_j}\\ &= \sum_i(\alpha_i-\beta_i)(-\sum_t \gamma_t^i(j)\Sigma_j^{-1}(x_t-\mu_j))\\ &+ D_j(\Sigma_j^{-1}(\mu_j-\mu_j^0))\ . &(6)\end{aligned}$$

By setting them to zero, it implies,

$$\alpha_i+\beta_i = 1 \qquad \forall i \tag{7}$$

$$\mu_j = \Phi_j(\alpha,\beta) = \frac{\sum_i(\alpha_i-\beta_i)\sum_t \gamma_t^i(j)x_t + D_j\mu_j^0}{\sum_i(\alpha_i-\beta_i)\sum_t\gamma_t^i(j)+D_j}\ , \tag{8}$$

and this is the GBW update equation for mean vectors.

If the optimization is performed on the covariance, the modification to the optimization problem is

$$\begin{aligned}\min_{\epsilon,\Sigma}\sum_i\epsilon_i + \sum_j\frac{D_j}{2}(\mu_j^{0'}\Sigma_j^{-1}\mu_j^0 + \mathsf{tr}(\Sigma_j^0\Sigma_j^{-1}) + \log|\Sigma_j|)\\ \text{s.t.}\quad \epsilon_i \geq Q_i(\Sigma)-C_i \quad \forall i\\ \epsilon_i \geq C_i - Q_i(\Sigma) \quad \forall i\ ,\end{aligned} \tag{9}$$

where Σ_j^0 is the covariance that we want GBW to backoff to. Similar to the optimization problem for solving the mean vectors, we setup the Lagrangian,

$$\begin{aligned}L_c(\epsilon,\Sigma,\alpha,\beta) &= \sum_i\epsilon_i - \sum_i\alpha_i(\epsilon_i - Q_i(\Sigma)+C_i)\\ &- \sum_i\beta_i(\epsilon_i - C_i + Q_i(\Sigma))\\ &+ \sum_j\frac{D_j}{2}(\mu_j^{0'}\Sigma_j^{-1}\mu_j^0 + \mathsf{tr}(\Sigma_j^0\Sigma_j^{-1})\\ &+ \log|\Sigma_j|)\ . &(10)\end{aligned}$$

We then differentiate the L_c w.r.t. the covariance,

$$\begin{aligned}\frac{\partial L_c}{\partial \Sigma_j} &= \sum_i(\alpha_i-\beta_i)\sum_t\gamma_t^i(j)(\Sigma_j^{-1}-\Sigma_j^{-1}S_{tj}\Sigma_j^{-1})\\ &+ D_j(\Sigma_j^{-1}-\Sigma_j^{-1}\Sigma_j^0\Sigma_j^{-1}-\Sigma_j^{-1}\mu_j^0\mu_j^{0'}\Sigma_j^{-1}) &(11)\end{aligned}$$

where $S_{tj}\equiv(x_t-\mu_j)(x_t-\mu_j)'$. Then by setting it to zero, we obtain the GBW update equation for covariance,

$$\begin{aligned}\Sigma_j &= \Psi_j(\alpha,\beta)\\ &= \frac{\sum_i(\alpha_i-\beta_i)\sum_t\gamma_t^i(j)x_tx_t' + D_j(\Sigma_j^0+\mu_j^0\mu_j^{0'})}{\sum_i(\alpha_i-\beta_i)\sum_t\gamma_t^i(j)+D_j} - \mu_j\mu_j'\ , &(12)\end{aligned}$$

The GBW update equations are generalization of BW and EBW update equations. GBW reduces to BW if $\alpha_i = 1$ and $\beta_i = 0$ for all references, $\alpha_i = \beta_i = 0.5$ for all competitors and $D_j = 0$. GBW is also equivalent to EBW if $\alpha_i = 1$ and $\beta_i = 0$ for all references, and $\alpha_i = 0$ and $\beta_i = 1$ for all competitors. Hence, GBW is a generalization of BW and EBW. In practice, the α_i and β_i are determined by solving a convex dual problem and details are available in [3].

3. Cross Entropy and Regularization

The GBW algorithm gives an interesting insight about the EBW algorithm. It states that the D-term in the EBW algorithm comes from some distance based regularization. In fact, GBW further explains that such regularization is based on a well known similarity measure between two probability distributions, i.e. KL divergence.

If we combine the optimization problems for solving mean vectors and covariance matrices into one single problem, we have,

$$\begin{aligned}\min_{\epsilon,\mu,\Sigma}\sum_i\epsilon_i + \sum_j\frac{D_j}{2}(||\mu_j-\mu_j^0||_{\Sigma_j} + \mathsf{tr}(\Sigma_j^0\Sigma_j^{-1}) + \log|\Sigma_j|)\\ \text{s.t.}\quad \epsilon_i \geq Q_i(\mu,\Sigma)-C_i \quad \forall i\\ \epsilon_i \geq C_i - Q_i(\mu,\Sigma) \quad \forall i\ .\end{aligned} \tag{13}$$

The regularization function is the KL-divergence from $N_0(\mu_j^0,\Sigma_j^0)$ to $N(\mu_j,\Sigma_j)$. If we put back the terms that are removed by differentiation,

$$\begin{aligned}\mathsf{KL}(N_0||N) &= \frac{1}{2}[||\mu_j-\mu_j^0||_{\Sigma_j} + \mathsf{tr}(\Sigma_j^0\Sigma_j^{-1})\\ &- \log\frac{|\Sigma_j^0|}{|\Sigma_j|} - K]\ , &(14)\end{aligned}$$

where K is the dimension of the feature vector. It is important to note that the term $\mu_j^0\Sigma_j^{-1}\mu_j^0$ is moved from the mean optimization problem to the covariance optimization problem. This term is part of the Mahalanobis distance but it disappears when we differentiate the objective function with respect to the mean vectors, hence, it remains in the covariance problem as shown in equation 9.

Equation 13 and 14 show that the D-term in the EBW update equation comes from the KL-divergence. Without affecting the solution of the optimization problem, we use cross entropy as the regularization function,

$$\mathsf{CH}(N_0||N) = H(N_0) + \mathsf{KL}(N_0||N)\ . \tag{15}$$

This does not alter the solution because the entropy of the backoff Gaussian N_0,

$$H(N_0) = \frac{1}{2}\log((2\pi e)^K|\Sigma_0|)\ , \tag{16}$$

is not related to the mean and covariance that we are optimizing. The function $H(N_0)$ is derived from differential entropy and details are available in [4].

In this setting, cross entropy measures the average number of bits required to encode N given N_0 is the true distribution. This is reasonable for regularization since cross entropy increases when N moves too far away from the backoff Gaussian N_0. However, N_0 in the EBW algorithm is either the ML model or the model from the previous EM iteration. In most cases, N_0 is inferior and it is not the true distribution. While the true distribution is unknown, we can look for a better Gaussian for the backoff purpose.

In this paper, we suggest we can treat the EBW/GBW update equations as some recurrence relations. The M-step of the EBW algorithm becomes an iterative procedure,

$$\mu_j^{m+1} = \frac{\sum_i(\alpha_i-\beta_i)\sum_t\gamma_t^i(j)x_t + D_j\mu_j^m}{\sum_i(\alpha_i-\beta_i)\sum_t\gamma_t^i(j)+D_j}\ , \tag{17}$$

$$\Sigma_j^{m+1} = \frac{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j)x_t x_t' + D_j(\Sigma_j^m + \mu_j^m \mu_j^{m'})}{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) + D_j} - \mu_j^{m+1}\mu_j^{m+1'}, \quad (18)$$

where μ_j^{m+1} and Σ_j^{m+1} are the Gaussian parameters of the $(m+1)$-th iteration, which depend on the parameters on the m-th iteration; If we perform only one iteration, it is the same the standard EBW/GBW algorithm. If we perform two iterations, it is like we are using the Gaussian computed from standard EBW/GBW algorithm as a backoff parameter. If we believe the Gaussian computed from the standard EBW/GBW algorithm is better than the original model, we are using a better estimate to compute the cross entropy for regularization. In this paper, we use the variable M to denote how many M-steps are performed after each E-step.

The reason for choosing cross entropy instead of KL-divergence is to examine the convergence of this recurrence relation, and whether the recurrence update leads to a smaller cross entropy. One can compare the cross entropy of successive iterations since it is measured by the number of bits. KL-divergence is a relative measure and it cannot compare the results of different iterations. Although equation 17 and equation 18 form simple linear recurrence relation, it is still impossible to prove convergence unless one can derive a bound on the feature vectors, x_t. In practice, we found that the cross entropy always decreases which implies the changes on the Gaussian parameters diminish across iterations. Details on this are available in section 4.

We would like to emphasize that the implementation of the above recurrence update equations is very simple. One can perform multiple M-steps in the standard EBW/GBW algorithm to achieve the same result. This incurs negligible extra computation since the M-step does not involve data processing. In this paper, we focus on the effectiveness of this new EBW algorithm. Hence, we do not test the recurrence GBW algorithm, but simply use GBW as a tool to derive this new EBW algorithm.

4. Experimental Setup

We evaluated the performance of the proposed EBW algorithm on three systems. The experiments included a Farsi ASR system, an Iraqi ASR system and a modern standard Arabic (MSA) ASR system. Table 1 summaries the configuration of these three systems. This table also contains the time needed for each EM iteration of the EBW algorithm. The time was measured by using 10 cores running in parallel and each core had similar performance to the Intel Xeon X5355 series at 2.66GHz. It demonstrated discriminative training is very expensive. Detailed system description of the Farsi and Iraqi ASR is available in [5] and description of the MSA ASR system is available in [6].

	Farsi ASR	Iraqi ASR	MSA ASR
Train data	110 hr	450 hr	1100 hr
System type	SI, 1-pass	SA, 1-pass	SA, 3-pass
Vocab size	33K	62K	737K
Adaptation	None	Incremental	Batch
# Gaussians	112K	308K	867K
LM	3-gram	3-gram	4-gram
Time/Iter	$\sim$ 3 hours	$\sim$ 20 hours	$\sim$ 10 days

Table 1: Description of the Farsi, Iraqi and MSA ASR systems.

For the experiments, the Farsi system used the TransTac Jul07 Farsi open set as the unseen test set. The Iraqi system used the TransTac Jun08 open set as dev set, and Nov08 open set as the unseen test set. The MSA system used GALE dev07/08/09 as dev sets, and eval09 and a three hours subset of dev10 as the unseen test sets.

We first investigated how the recurrence update equations affect the performance of the new EBW algorithm. We compared the EBW algorithm with different number of M-steps per EM iteration using the recurrence equation 17 and 18. Both EBW algorithms optimize the acoustic model for the BMMI objective function. We used the Iraqi system to analyze the performance. In this experiment, We tried up to four EM iterations and for each EM iteration, we performed a fixed number of M-steps from one to four ($M = 1, 2, 3, 4$).

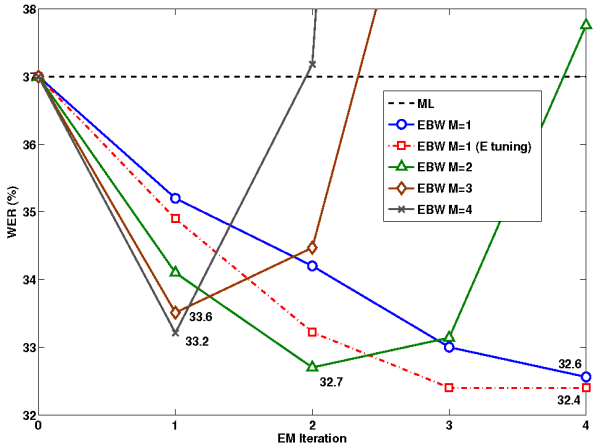

Figure 1: Performance of EBW algorithm with different number of M-steps per EM iteration. This experiment is performed on the TransTac Jun08 open set using the Iraqi ASR system.

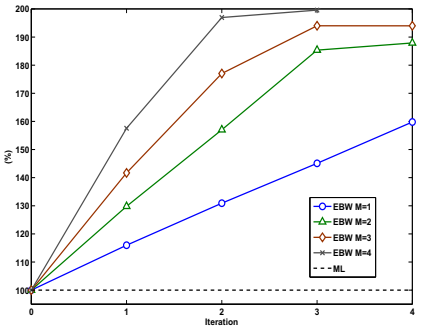

Figure 2: Increase of the BMMI objective function compared to the BMMI score of the ML model on the train set.

Figure 1 shows that if we perform more M-steps per EM iteration, the system can achieve the best performance at earlier iterations. However, as shown in figure 2, performing multiple M-steps may also cause overfitting to occur earlier than the standard EBW algorithm as the training becomes more aggressive. When we perform two M-steps per EM iteration ($M = 2$), we got 32.7% WER which is almost the same as the 32.6% WER of standard EBW ($M = 1$) with only half the training time. We also tried the standard EBW algorithm with a grid search of learning rate (E tuning). In the model update equation 8 and 12, D_j controls the weight of the regularization. This value is often computed by a heuristics and it is the maximum of $E \times \sum_t \gamma_t^c(j)$, or twice the value required to keep the covariance positive. E is often set to two and it is also our set-

ting for all EBW algorithms except the one with grid search. The grid search is performed based on the WER of the test set, which we find the best E in the range $[1.0, 3.0]$. Therefore, it is an oracle experiment. The purpose of this oracle experiment is to investigate if the standard EBW algorithm, in the optimal case, can converge as fast as our proposed EBW algorithm. Our results showed the opposite, and it implied our method is useful. Figure 3 shows the reduction in average cross entropy for

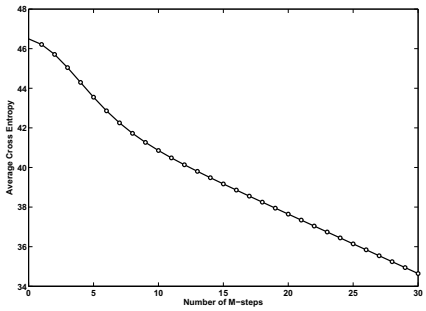

Figure 3: Decrease in average cross entropy implies the changes on the Gaussian parameters diminish for each M-step.

each M-step performed. The cross entropy is computed after the first EM iteration shown in figure 1 and it is averaged across all Gaussian distributions in the acoustic model. This result shows that the cross entropy is decreasing so it implies the changes in the Gaussian parameters are also decreasing.

Based on these results, we studied whether our proposed EBW algorithm causes accuracy degradation as a tradeoff for faster convergence. We compared the performance of the new EBW algorithm with the standard version on our Farsi ASR, Iraqi ASR and MSA ASR systems. In this experiment, the new EBW algorithm performed two M-steps for each E step ($M = 2$). In total, two EM iterations were performed. The standard EBW algorithm performed four EM iterations and one M-step per E-step ($M = 1$). Therefore, the execution time of the new EBW algorithm is only half of the standard version. Table 2, 3 and 4 showed the performance of the Farsi, Iraqi and MSA ASR systems respectively.

	Farsi Jul07 open
BW_{ML}	50.2%
$EBW_{M=1}$	45.6%
$EBW_{M=2}$	45.5%

Table 2: The WER of the Farsi ASR system on the Jul07 open set.

	Jun08 open	Nov08 open
BW_{ML}	37.0%	35.2%
$EBW_{M=1}$	32.6%	30.6%
$EBW_{M=2}$	32.7%	30.8%

Table 3: The WER of the Iraqi ASR system on the Jun08 and Nov08 open sets.

The results suggested that our proposed EBW algorithm can achieve the same WER as the standard EBW algorithm. Among these eight test sets on three different systems, the difference in WER is no more than 0.2% absolute. Therefore, the gain in speed is a clear advantage for the new EBW algorithm. According to the information in table 1, using the standard EBW algorithm needs 40 days to train the MSA system, while the new EBW algorithm only needs around 20 days to achieve the same WER, which is a big advantage.

	dev07	dev08	dev09	eval09	dev10
BW_{ML}	13.7%	15.5%	20.4%	15.1%	16.5%
$EBW_{M=1}$	11.7%	14.0%	18.6%	13.3%	14.6%
$EBW_{M=2}$	11.9%	14.0%	18.5%	13.2%	14.5%

Table 4: The WER of the MSA ASR system on the GALE dev07/08/09/10 and eval09 test sets.

5. Conclusion and Future Work

We demonstrated how to use the GBW algorithm to develop a better EBW algorithm. The GBW algorithm showed that the D-term of the EBW algorithm came from the KL-divergence/cross entropy. Based on this information, we proposed a fast EBW algorithm which can cut the time of model space discriminative training by half, without performance loss. In sum, the GBW algorithm allows us to understand the EBW algorithm better, and hence, we can improve it.

There are other ways to develop variants of the EBW algorithm. Instead of using one Gaussian model as a backoff model to compute cross entropy, one can use multiple models from the same HMM state and create multiple regularization terms in the optimization problem. We will investigate this variant of the EBW algorithm in the future.

6. Acknowledgments

This work is in part supported by US DARPA under the TransTac (Spoken Language Communication and Translation System for Tactical Use) program, and the GALE (Global Autonomous Language Exploitation) program under Contract No. HR0011-06-2-0001. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

7. References

[1] Y. Normandin and S. D. Morgera, " An Improved MMIE Training Algorithm for Speaker-independent, Small Vocabulary, Continuous Speech Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1991.

[2] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for Model and Feature-space Discriminative Training," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 4057–4060.

[3] R. Hsiao, Y.C. Tam, and T. Schultz, "Generalized Baum-Welch Algorithm for Discriminative Training on Large Vocabulary Continuous Speech Recognition System," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2009.

[4] N.A. Ahmed and D.V. Gokhale, "Entropy Expressions and their Estimators for Multivariate Distributions," *IEEE Transactions on Information Theory*, vol. 35, pp. 688–692, 1989.

[5] N. Bach, M. Eck, P. Charoenpornsawat, T. Köhler, S. Stüker, T. Nguyen, R. Hsiao, A. Waibel, S. Vogel, T. Schultz, and A. W. Black, "The CMU TransTac 2007 Eyes-free, and Hands-free Two-way Speech-to-speech Translation System," in *Proceedings of the IWSLT*, 2007.

[6] F. Metze, R. Hsiao, Q. Jin, U. Nallasamy, and T. Schultz, "The 2010 CMU GALE Speech-to-Text System," in *Proceedings of the INTERSPEECH*, Makuhari, Japan, 2010.