GENERALIZED BAUM-WELCH ALGORITHM FOR DISCRIMINATIVE TRAINING ON LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION SYSTEM

Roger Hsiao, Yik-Cheung Tam and Tanja Schultz

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

ABSTRACT

We propose a new optimization algorithm called Generalized Baum Welch (GBW) algorithm for discriminative training on hidden Markov model (HMM). GBW is based on Lagrange relaxation on a transformed optimization problem. We show that both Baum-Welch (BW) algorithm for ML estimate of HMM parameters, and the popular extended Baum-Welch (EBW) algorithm for discriminative training are special cases of GBW. We compare the performance of GBW and EBW for Farsi large vocabulary continuous speech recognition (LVCSR).

***Index Terms*—** Speech recognition, discriminative training.

1. INTRODUCTION

Discriminative training is an important technique to improve recognition accuracy for large vocabulary continuous speech recognition (LVCSR) [1][2]. Common discriminative training algorithms in speech recognition employ maximum mutual information (MMI) estimation [1] and minimum phone error (MPE) [2]. While MMI and MPE have different objective functions, they use the the extended Baum-Welch (EBW) algorithm [3] for optimization. In recent years, large margin based approach gains popularity such as [4] which shows promising results. However, on large scale system, approaches based on lattices like MMI/MPE using EBW algorithm remain to be the most popular methods.

One of the major challenges of discriminative training is optimization. The objective functions used in discriminative training like MMI/MPE can be unbounded. It is also the reason why EBW may corrupt the acoustic model if it is not properly tuned and smoothed. In this paper, we propose a new optimization algorithm called Generalized Baum Welch (GBW). GBW is based on Lagrange relaxation [5] and the optimization is operated in a dual space. We show that GBW does not have the unbound issue and it does not corrupt the model due to improper tuning. More importantly, we show that both Baum-Welch (BW) algorithm for maximum likelihood (ML) estimate, and EBW for MMI/MPE estimate are special cases of GBW. The formulation of GBW also gives us a new insight of EBW formulation which is naturally understood in GBW framework.

This paper is organized as follows: in section 2, we review the EBW algorithm and the MMI objective. In section 3, we formulate the GBW algorithm to generalize BW and EBW algorithm. In section 4, we report experimental results on EBW and GBW. We conclude our work and discuss future work in section 5.

2. EXTENDED BAUM-WELCH ALGORITHM

The objective function for discriminative training, in its simplest form, involves the difference between two log likelihood functions. Consider the simplest case that we only have one reference and one competitor, then,

$$F(X, \theta) = Q_r(X, \theta) - Q_c(X, \theta) \ , \qquad (1)$$

where $Q = \sum_t \sum_j \gamma_t(j)[\log |\Sigma_j| + (x_t - \mu_j)'\Sigma_j^{-1}(x_t - \mu_j)]$ is an auxiliary function to represent the negative log likelihood. In which, x is the observation; $\gamma_t(j)$ is the posterior probability of x being at Gaussian j at time t. The function F represents the difference between the negative log likelihood of the reference, Q_r, and the competitor Q_c on observation $X = \{x_1, \dots, x_T\}$. θ is the model parameter set including the mean vectors (μ), covariances (Σ) and mixture weights in a HMM. Minimization of F is the same as maximizing the mutual information so this form of discriminative training is also known as MMI estimation. MPE is based on the same principle but it has a more sophisticated objective function.

The function F is non-convex and optimization of F can be local optimal. Another bigger problem of F is the unbounded issue. For example, if a Gaussian appears only as a competitor, optimizing the parameters of this Gaussian becomes a minimum likelihood problem, which is unbounded. In general, if the denominator occupancy of a Gaussian is higher than its numerator occupancy, the solution is unbounded. In sum, optimization of F is not trivial.

The idea of EBW is to add an additional auxiliary function to the function F to enforce convexity. That auxiliary function is required to have zero gradient at the current parameter set [2]. Details of EBW is available in [3] and the reestimation formula for mean μ_j is:

$$\mu_j = \frac{\sum_t \gamma_t^r(j)x_t - \sum_t \gamma_t^c(j)x_t + D_j\mu_j^0}{\sum_t \gamma_t^r(j) - \sum_t \gamma_t^c(j) + D_j} \ , \qquad (2)$$

where the subscript shows whether the term belongs to reference (r) or competitor (c); D_j is a constant chosen to guarantee the estimate is valid (say, covariance must be positive definite). Comparing to the Baum Welch algorithm which provides ML estimate for HMM, EBW algorithm considers the competitors as well. [3] shows that EBW algorithm converges when $D \to \infty$, where D is directly proportional to the number of discrete distributions used to represent a Gaussian in continuous space. It is the reason why EBW needs $D \to \infty$ to guarantee convergence.

In practice, we cannot choose $D \to \infty$, so EBW is not guaranteed to converge. In addition, the EBW reestimation formula in equation 2 often leads to overtraining. Hence, smoothing technique such as I-smoothing [2], has been proposed. While EBW has been proven successful in practice, it often involves careful tuning.

3. GENERALIZED BAUM-WELCH ALGORITHM

We introduce Generalized Baum Welch (GBW) algorithm in this section. GBW algorithm uses Lagrangian relaxation on a transformed optimization problem, and we optimize the parameters for the dual problem which itself is a relaxed problem . GBW does not have the unbounded issue and we can show that both BW and EBW are special cases of GBW.

3.1. Checkpointing

As mentioned, optimizing the function F can be unbounded to some parameters. However, we can address the unbounded issue by adding checkpoints to the problem,

$$G(X,\theta) = |Q_r(X,\theta) - C_r| + |Q_c(X,\theta) - C_c| \qquad (3)$$

where C_r and C_c are the checkpoints that we want Q_r and Q_c to achieve respectively. For this particular example, we choose the checkpoints such that $Q_r(X,\theta) > C_r$ and $Q_c(X,\theta) < C_c$. As a result, by minimizing the function G, we are maximizing the log likelihood difference between the reference and the competitor, but we only want them to achieve the checkpoints we have chosen. In general, we have multiple files and each file has possibly multiple competitors. Hence, the formulation becomes,

$$G(X,\theta) = \sum_i |Q_i(X,\theta) - C_i| \; . \qquad (4)$$

Note that this formulation is very flexible that we can represent reference and competitors at different granularity levels. Since we are using a lattice based approach, each term in equation 4 corresponds to a word arc. As a result, we have multiple terms for reference and competing word arcs and each word arc has its own checkpoint. It is also important to note that when each term corresponds to a word arc, not every term has equal importance because of different posterior count. To reflect this, one may add a weighting factor for each term or scale the checkpoints. The formulas shown in this paper, however, assume they have equal importance for simplicity, but it is trivial to incorporate this information into the algorithm.

Although the function G remains to be non-convex, this formulation has an obvious advantage over the original problem. The reason is the unbounded issue no longer exists in this form, since G must be larger than or equal to zero. One easy way to define the checkpoints is to encourage higher likelihood for the reference word arcs and lower likelihood for the competing word arcs. This scheme is equivalent to MMI estimation.

3.2. Lagrange Relaxation

Assuming good checkpoints are given so that if our model can reach those checkpoints, the model can achieve good performance. To minimize the function G, we may first transform the problem to,

$$\begin{array}{ll} \min\limits_{\epsilon,\theta} & \sum_i \epsilon_i \\ \text{s.t.} & \epsilon_i \geq Q_i(\theta) - C_i \quad \forall i \\ & \epsilon_i \geq C_i - Q_i(\theta) \quad \forall i \; , \end{array}$$

where ϵ represents slack variables and i is an index to a word arc. This is equivalent to the original problem in equation 4 without constraints. We call this as the primal problem for the rest of this paper.

For simplicity, we show the formulation for optimizing the mean vectors, and this formulation also includes an optional regularization using Mahalanobis distance on the means. We would like to emphasize that this method also allows us to train covariances and the optional regularization is not required for GBW to work. The primal problem becomes,

$$\begin{array}{lll} \min\limits_{\epsilon,\mu} & \sum_i \epsilon_i & + \sum\limits_j D_j ||\mu_j - \mu_j^0||_{\Sigma_j} \\ \text{s.t.} & \epsilon_i \geq Q_i(\mu) - C_i \quad \forall i & \\ & \epsilon_i \geq C_i - Q_i(\mu) \quad \forall i \; , & \end{array} \qquad (5)$$

where D_j is a Gaussian specific constant to control the importance of the regularization term; μ_j^0 is the mean vector that we want GBW to backoff to, and it is assumed to be an ML estimate here.

We can then construct the Lagrangian dual for the primal problem. The Lagrangian is defined as,

$$\begin{aligned} L_m(\epsilon,\mu,\alpha,\beta) &= \sum_i \epsilon_i - \sum_i \alpha_i(\epsilon_i - Q_i(\mu) + C_i) \\ &- \sum_i \beta_i(\epsilon_i - C_i + Q_i(\mu)) \\ &+ \sum_j D_j ||\mu_j - \mu_j^0||_{\Sigma_j} \end{aligned} \qquad (6)$$

where $\{\alpha_i\}$ and $\{\beta_i\}$ are the Lagrange multipliers for the first and the second set of constraints of the primal problem in equation 5. The Lagrangian dual is then defined as,

$$L^D(\alpha,\beta) = \inf_{\epsilon,\mu} L_m(\epsilon,\mu,\alpha,\beta) \qquad (7)$$

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

$$\begin{aligned} \frac{\partial L_m}{\partial \epsilon_i} &= 1 - \alpha_i - \beta_i \qquad (8) \\ \frac{\partial L_m}{\partial \mu_j} &= \sum_i (\alpha_i - \beta_i) \frac{\partial Q_i}{\partial \mu_j} + D_j \frac{\partial}{\partial \mu_j} ||\mu_j - \mu_j^0||_{\Sigma_j} \\ &= \sum_i (\alpha_i - \beta_i)(-2\sum_t \gamma_t^i(j) \Sigma_j^{-1}(x_t^i - \mu_j)) \\ &+ D_j(2\Sigma_j^{-1}(\mu_j - \mu_j^0)) \; . \qquad (9) \end{aligned}$$

By setting them to zero, it implies,

$$\alpha_i + \beta_i = 1 \quad \forall i \qquad (10)$$

and,

$$\mu_j = \Phi_j(\alpha,\beta) = \frac{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) x_t^i + D_j \mu_j^0}{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) + D_j} \; , \qquad (11)$$

and this is the GBW update equation for mean vectors.

BW algorithm is a special case of GBW, since if we disable the regularization ($D = 0$) and set all α to one and β to zero for reference word arcs and $\alpha = \beta = 0.5$ for all competitors, we get

$$\mu_j = \frac{\sum_{i \in \text{ref}} \sum_t \gamma_t^i(j) x_t^i}{\sum_{i \in \text{ref}} \sum_t \gamma_t^i(j)} \; , \qquad (12)$$

which is the BW update equation. EBW is also a special case of GBW, since if we set α equals one and β equals zero for all reference, and α equals zero and β equals one for all competitors, the GBW update equation becomes EBW update equation,

$$\mu_j = \frac{\sum_{i \in \text{ref}} \sum_t \gamma_t^i(j) x_t^i - \sum_{i \in \text{com}} \sum_t \gamma_t^i(j) x_t^i + D_j \mu_j^0}{\sum_{i \in \text{ref}} \sum_t \gamma_t^i(j) - \sum_{i \in \text{com}} \sum_t \gamma_t^i(j) + D_j} \; . \qquad (13)$$

One should note that this result implies the D-term used in EBW can be considered as a regularization using Mahalanobis distance between the mean vectors of the new and the ML model, and the meaning is well represented.

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

$$\begin{aligned}\min_{\Sigma} \sum_i \epsilon_i + \sum_j D_j(\mathsf{tr}(A'_j \Sigma_j^{-1} A_j) + \mathsf{tr}(B'_j \Sigma_j^{-1} B_j) + \log|\Sigma_j|) \\ \text{s.t.} \quad \epsilon_i \geq Q_i(\Sigma) - C_i \quad \forall i \\ \epsilon_i \geq C_i - Q_i(\Sigma) \quad \forall i \ , \qquad (14)\end{aligned}$$

where $\Sigma_j^0 = A'_j A_j$; $M_j^0 \equiv \mu_j^0 \mu_j^{0'} = B'_j B_j$. Assuming both A and B exist, we have this Lagrangian, L_c,

$$\begin{aligned} L_c(\epsilon, \Sigma, \alpha, \beta) &= \sum_i \epsilon_i - \sum_i \alpha_i(\epsilon_i - Q_i(\Sigma) + C_i) \\ &- \sum_i \beta_i(\epsilon_i - C_i + Q_i(\Sigma)) \\ &+ \sum_j D_j(\mathsf{tr}(A'_j \Sigma_j^{-1} A_j) + \mathsf{tr}(B'_j \Sigma_j^{-1} B_j) \\ &+ \log|\Sigma_j|) \ . \qquad (15)\end{aligned}$$

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

$$\begin{aligned}\frac{\partial L_c}{\partial \Sigma_j} &= \sum_i (\alpha_i - \beta_i) \sum_t \gamma_t^i(j)(\Sigma_j^{-1} - \Sigma_j^{-1} S_{tj}^i \Sigma_j^{-1}) \\ &+ D_j(\Sigma_j^{-1} - \Sigma_j^{-1}\Sigma_j^0\Sigma_j^{-1} - \Sigma_j^{-1} M_j^0 \Sigma_j^{-1}) \ , \qquad (16)\end{aligned}$$

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

$$\begin{aligned}\Sigma_j &= \Psi_j(\alpha, \beta) \\ &= \frac{\sum_i (\alpha_i - \beta_i) \sum_t \gamma_t^i(j)(x_t^i - \mu_j)(x_t^i - \mu_j)' + D_j(\Sigma_j^0 + M_j^0)}{\sum_i (\alpha_i - \beta_i) \sum_t \gamma_t^i(j) + D_j} \ , \qquad (17)\end{aligned}$$

which is also a generalization of BW and EBW. Instead of solving two independent optimization problems, one may use the parameters obtained from the first problem as the solution for the second problem to compute the covariances. This procedure assumes the solutions of the two problems are similar and we adopt this procedure in our experiments. One should also note that the formulation of GBW can incorporate I-smoothing [2] easily as well.

GBW is the same as BW and EBW that it is based on the EM algorithm. However, the M-step of GBW is replaced by solving a dual problem to retrieve the Lagrange multipliers, so we can use equation 11 and equation 17 to obtain the HMM parameters. The dual problem is formulated by plugging equations 10, 11 and 17 back to the Lagrangian. Assuming we are optimizing the mean vectors, we have

$$\begin{aligned}\max_{\alpha,\beta} \quad L^D(\alpha, \beta) &= \sum_i (\alpha_i - \beta_i)(Q_i(\Phi(\alpha, \beta)) - C_i) \\ \text{s.t. } \forall i \quad \alpha_i + \beta_i &= 1 \text{ and } \alpha_i, \beta_i \geq 0 \ .\end{aligned}$$

This dual problem can be solved by gradient ascent. By taking derivative w.r.t. the Lagrange multipliers, we obtain the gradients. We need an assumption that at each iteration, the parameters do not move too far away. If this assumption holds, we can assume the denominators of equation 11 and 17 are unchanged. Otherwise, the gradient equation would couple with all the multipliers in the program which would become computationally intractable. Finally, we have,

$$\frac{\partial L^D}{\partial \alpha_i} = Q_i - C_i + \sum_j (\alpha_j - \beta_j) \sum_k \frac{\partial Q_j}{\partial \Phi_k} \frac{\partial \Phi_k}{\partial \alpha_i} \ , \qquad (18)$$

and,

$$\frac{\partial \Phi_k}{\partial \alpha_i} = \frac{\sum_t \gamma_t^i(k) x_t^i}{Z_k(\alpha, \beta)} \qquad (19)$$

where $Z_k(\alpha, \beta) = \sum_i (\alpha_i - \beta_i) \sum_t \gamma_t^i(k) + D_k$ and it is considered to be constant and we can obtain this value from the past iteration. When α_i is updated, β_i can be obtained using the constraint $\alpha_i + \beta_i = 1$.

3.3. Convergence condition of EBW and GBW

This technique we use for GBW is known as Lagrange relaxation [5], since it converts a primal problem into a dual problem. In theory, the dual problem is always a convex problem (maximizing a concave objective function here) [5]. Note that when strong duality does not hold, which means the optimal value of the dual can only serve as a strict lower bound to the primal objective, there is no guarantee that the solution obtained from the dual is primal optimal. We can only consider this technique as a relaxation method.

Consider when $D \to \infty$ and this term dominates the objective function, strong duality occurs and GBW is guaranteed to converge in this case. Although the solution is simply the backoff model, this behavior is the same as EBW. However, given a problem and a finite D, if the solution of GBW is equivalent to BW or EBW, it can be shown GBW is guaranteed to converge for this specific problem. One should also note that the D constant in GBW is related to the checkpoints. If the checkpoints are set more aggressively, that is very high likelihood for reference word arcs and very low likelihood for competing word arcs, GBW is very likely to reduce to EBW (but it is possible to construct artificial cases that GBW does not reduce to EBW). However, in such case, the ϵ of the primal problem becomes larger, and therefore, D has to be larger for regularization to be effective. Hence, although we claim GBW must converge when it reduces to EBW, this case is equivalent to having $D \to \infty$.

4. EXPERIMENTAL SETUP

We evaluated the performance of GBW and EBW on a speaker independent Farsi LVCSR system with 33K vocabulary. The Farsi system was trained with more than 110 hours of audio data in force protection and medical screening domain. The audio data can be roughly divided into two categories: 1.5-way and 2-way data. 1.5-way means basic question and answering and the sentences tend to be simpler; 2-way data is conversational and it may have more complicated or incomplete sentences. A development test set was selected from the 2-way data set as we are interested in conversational data. This development set consists of around 45 minutes of 2-way data. For the test set, we selected the Farsi offline evaluation set used in DARPA TransTac 2007 evaluation, which consists of around 2 hours of conversational data. We tuned the algorithms based on the development set and tested on the test set at the end.

MMI objective was chosen for optimization. The checkpoints were selected based on the model used in E-step, and they were set to be 10% to 40% higher than the log likelihood of the reference word arcs, and 10% to 40% lower of the competing word arcs. In the M-step, we performed four iterations of gradient ascent to update the dual variables. From the dual variables, we then reestimated the Gaussian parameters. No regularization nor smoothing was used for GBW in the first experiment.

The result in figure 1 shows that GBW without regularization and smoothing can improve the baseline ML system. It shows GBW

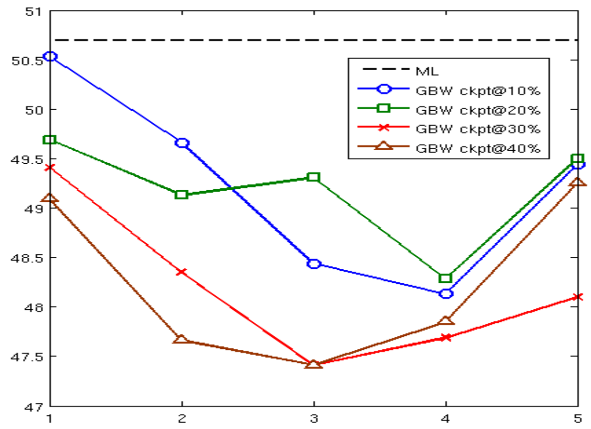

Fig. 1. Performance of GBW without regularization on dev set.

is reliable as it works even without regularization and smoothing. On the contrary, EBW does not work when there is no regularization or smoothing and it just corrupts the model. As the checkpoints are set more aggressively, GBW gives more improvement at earlier iterations but degrades afterwards.

Model initialization for GBW is important due to the EM framework, one option is to initialize the dual variables such that it conforms to the ML model, that is BW initialization. Another option is to initialize the dual variables with EBW after the first iteration of EBW. Figure 2 shows the performance of GBW and EBW with different settings. Although the figure only shows the first seven iterations, the experiment was performed with 16 iterations and no improvement after the first seven iterations is observed for all algorithms.

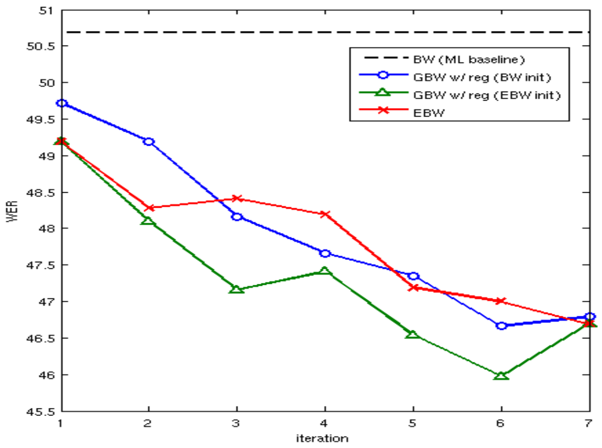

Fig. 2. Performance of BW, EBW and GBW on dev set.

When GBW is initialized as EBW, GBW outperforms EBW at all iterations. GBW with BW initialization lags behind EBW at the earlier stages of the training since GBW is close to ML at the beginning, but GBW can obtain the same performance of EBW at the end. When BW initialization is used, in addition to the figure, GBW without regularization and smoothing gives more improvement at the early stages compared to the one with regularization. However, it over trains the system very soon due to the aggressiveness when regularization is not used.

Table 1 summarizes the WER performance of EBW and GBW on the test set. Both EBW and GBW make significant improvement over the baseline ML model and GBW performs slightly better than EBW.

algo	obj func	dev	test
BW	ML	50.7%	50.2%
EBW	MMI	46.7%	46.5%
GBW	MMI	46.0%	45.8%

Table 1. WER of BW, EBW, and GBW on dev set and TransTac 2007 Farsi offline evaluation set.

5. CONCLUSION AND FUTURE WORK

We presented generalized Baum-Welch algorithm for discriminative training. We showed that the common BW and EBW algorithms are special cases of GBW. Unlike EBW, GBW uses a checkpointing technique to address the unbound issue, and GBW works even without regularization. Preliminary experiments also showed that GBW can improve BW and EBW algorithm. More experiments on the checkpoints, and the training procedure are needed in order to understand the behavior of this algorithm.

The formulation of GBW helps us to understand EBW better. We learn that the regularization and smoothing of EBW can be represented as a distance based regularization to the primal objective. Regularization and smoothing are not always necessary for GBW, but these methods improve the performance.

6. ACKNOWLEDGMENTS

This work is in part supported by the US DARPA under the TransTac (Spoken Language Communication and Translation System for Tactical Use) program. 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] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young, “MMIE Training of Large Vocabulary Recognition Systems,” *Speech Communication*, vol. 22, no. 4, pp. 303–314, 1997.

[2] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University Engineering Dept., 2003.

[3] 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.

[4] F. Sha and L. K. Saul, “Large Margin Hidden Markov Models for Automatic Speech Recognition,” *Advances in Neural Information Processing Systems*, vol. 19, pp. 1249–1256, 2007.

[5] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.