TOWARDS SINGLE PASS DISCRIMINATIVE TRAINING FOR SPEECH RECOGNITION

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 combine our previously proposed fast extended Baum-Welch algorithm and generalized discriminative feature transformation to achieve single pass discriminative training, which we only process the data once. Compared to the state of the art training procedure, which uses feature space maximum mutual information (fMMI) and boosted maximum mutual information (BMMI), our proposed training procedure can achieve around 80% of the improvement available from discriminative training. We also show that if we are allowed to process the data twice, it is possible to achieve almost all of the improvement. We evaluate different training procedures on various large scale tasks using Iraqi and modern standard Arabic speech recognition systems.

Index Terms— Speech recognition, discriminative training.

1. INTRODUCTION

Discriminative training is an expensive but effective process to improve recognition accuracy for automatic speech recognition (ASR) systems. The lengthy training time is often due to the huge amount of data required to build a high performance system. Also, as long as "there is no data like more data" remains true, one can foresee that discriminative training will dominate the development time for an ASR system. This is not desirable since the cost of discriminative training may eventually exceed the available processing power and it may hinder the researchers to exploit the virtually unlimited amount of data to improve an ASR system.

In our previous work, we proposed Generalized Baum-Welch (GBW) algorithm and Generalized Discriminative Feature Transformation (GDFT). Both algorithms give insights about the optimization involved in discriminative training. In [1], based on GBW, we proposed a variant of EBW algorithm which can reduce the training time by half without sacrificing any recognition performance. In [2], we proposed an optimization algorithm for joint feature space and model space discriminative training.

The goal of this paper is to combine our previous work and explore how much improvement we can achieve from discriminative training if we can process the data only once. If this single-pass discriminative training is feasible, it helps lowering the cost of discriminative training.

This paper is organized as follows: in section 2, we review the GBW algorithm and our proposed fast EBW algorithm. Then in section 3, we review GDFT and introduce some enhancement which can improve GDFT. In section 4, we report experimental results on the single-pass discriminative training. We conclude our work and discuss future work in section 5.

2. RECURSIVE EBW ALGORITHM

In [1], we proposed the GBW algorithm and we explained that both BW and EBW algorithms are special cases of our GBW algorithm. Instead of optimizing the discriminative objective function directly, GBW optimizes,

$$\min_{\mu,\Sigma} G(\mu,\Sigma) = \sum_i |Q_i(X,\mu,\Sigma) - C_i| + \sum_j D_j R(N_j, N_j^0) \quad (1)$$

where N_j is the j-th Gaussian distribution in the acoustic model; $X = x_1, \ldots, x_T$ represents the feature vectors; μ and Σ represent the mean vectors and the covariance matrices which we are optimizing; N_j^0 is the backoff Gaussian for N_j; i is an index referring to all the references and their competitors in the train set; Q_i is an auxiliary function representing negative log likelihood; C_i is the target value for Q_i and $R(N_j, N_j^0)$ is a regularization function.

When R is defined as the cross entropy from N_j^0 to N_j,

$$R(N_j, N_j^0) = \mathsf{CH}(N_j^0||N_j) = H(N_j^0) + \mathsf{KL}(N_j^0||N_j) \,, \quad (2)$$

where $H(N_j^0)$ is the entropy of the backoff Gaussian distribution,

$$H(N_j^0) = \frac{1}{2}\log((2\pi e)^K |\Sigma_j^0|) \,, \quad (3)$$

and $\mathsf{KL}(N_j^0||N_j)$ is the KL divergence from N_j^0 to N_j,

$$\begin{aligned} \mathsf{KL}(N_j^0||N_j) &= \frac{1}{2}[||\mu_j - \mu_j^0||^2_{\Sigma_j} + \mathsf{tr}(\Sigma_j^0 \Sigma_j^{-1}) \\ &- \log\frac{|\Sigma_j^0|}{|\Sigma_j|} - K] \,, \end{aligned} \quad (4)$$

with K meaning the dimension of the features, we obtain the update equations for the Gaussian distribution N_j,

$$\mu_j = \frac{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) x_t + D_j \mu^0}{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) + D_j} \,, \quad (5)$$

$$\begin{aligned} \Sigma_j = \frac{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) x_t x_t' + D_j(\Sigma_j^0 + \mu_j^0 \mu_j^{0'})}{\sum_i(\alpha_i - \beta_i)\sum_t \gamma_t^i(j) + D_j} \\ -\mu_j \mu_j \,, \end{aligned} \quad (6)$$

where α_i and β_i are the Lagrange multipliers used by the GBW algorithm [1].

Cross entropy measures the average number of bits required to encode N_j given N_j^0 is the true distribution. This is reasonable for

regularization since cross entropy increases when N_j moves too far away from the backoff Gaussian N_j^0. However, N_j^0 in the EBW algorithm is either the ML model or the model from the previous EM iteration. In most cases, N_j^0 is inferior and it is not the true distribution. While the true distribution is unknown, if we believe the model after the EBW update is better in terms of accuracy, we can use the updated model as the backoff model. By doing so, we treat the EBW/GBW update equations as some recurrence equations. The M-step of the EBW algorithm becomes an recursive procedure,

$$\mu_j^{m+1} = \frac{\sum_t \gamma_t^r(j)x_t - \sum_t \gamma_t^c(j)x_t + D_j\mu_j^m}{\sum_t \gamma_t^r(j) - \sum_t \gamma_t^c(j) + D_j} , \quad (7)$$

$$\Sigma_j^{m+1} = \frac{\sum_t \gamma_t^r(j)x_t x_t' - \sum_t \gamma_t^c(j)x_t x_t' + D_j(\Sigma_j^m + \mu_j^m \mu_j^{m'})}{\sum_t \gamma_t^r(j) - \sum_t \gamma_t^c(j) + D_j} - \mu_j^{m+1}\mu_j^{m+1'} , \quad (8)$$

where the subscripts r and c represent the reference/numerator and competitor/denominator statistics respectively; μ_j^{m+1} and Σ_j^{m+1} are the Gaussian parameters of the $(m+1)$-th iteration, which depend on the parameters of the m-th iteration; If we perform only one iteration, it is the same as 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. In this paper, we use the variable M to denote how many M-steps are performed after each E-step. In practice, we found that two to four M-steps is enough for faster convergence.

3. GENERALIZED DISCRIMINATIVE FEATURE TRANSFORMATION

This section investigates the possibility of performing single-pass feature space discriminative training. Feature space MMI/MPE (fMMI/MPE) [3] and region dependent linear transformation (RDLT) [4] are two commonly used algorithms for feature space discriminative training. Both methods use gradient based optimization to estimate the feature transforms. Each iteration of fMMI/MPE and RDLT requires three passes on the data. The first pass is to collect the statistics for computing the indirect gradient. The second pass computes the gradients and the final pass performs single-pass retraining or ML update of the acoustic models. Since fMMI/MPE and RDLT require multiple passes on the data for each iteration, they are expensive and not suitable to our scenario which only allows single-pass on the data. However, we still compare the performance of our proposed approaches with systems using fMMI and see how much gain can be obtained from feature space discriminative training.

In [2], we proposed generalized discriminative feature transformation(GDFT) to perform feature space discriminative training. GDFT can be considered as a discriminative version of constrained MLLR (CMLLR), which optimizes the feature transforms over the whole train set. GDFT uses the same mathematical framework of GBW which optimizes,

$$G(W) = \sum_i |Q_i(W) - C_i| + \frac{D}{2}||W - W^0||_F^2 , \quad (9)$$

where W is the linear transformation of GDFT with transform matrix A and bias b ($W \equiv [A; b]$); W^0 is the backoff transform which is either the identity transform or the transform from the previous EM iteration; $||W - W^0||_F^2$ is the Frobenius norm between W and W^0 and D is a tunable parameter controlling the weight of this regularization term.

Equation 9 can be solved by Lagrange relaxation like GBW, and GDFT has a similar update equations compared to CMLLR [5],

$$w_d = (\delta p_d + k^{(d)})G^{(d)-1} . \quad (10)$$

where w_d is the d-th row of W; $p_d = [c_{d1}, \ldots, c_{dn}, 0]$ is the extended cofactor row vector of A ($c_{ij} = cof(A_{ij})$) , and,

$$G^{(d)} = \sum_i (\alpha_i - \beta_i) \sum_j \frac{1}{\sigma_{jd}^2} \sum_t \gamma_t^i(j)\zeta_t\zeta_t' + DI \quad (11)$$

$$k^{(d)} = \sum_i (\alpha_i - \beta_i) \sum_j \frac{\mu_{jd}}{\sigma_{jd}^2} \sum_t \gamma_t^i(j)\zeta_t' + Dw_d^0 \quad (12)$$

$$\Gamma = \sum_i (\alpha_i - \beta_i) \sum_t \sum_j \gamma_t^i(j) , \quad (13)$$

where $\zeta_t = [x_t', 1]'$.

When GDFT is used with multiple transforms, GDFT is the same as fMMI/MPE and RDLT which uses a Gaussian mixture model (GMM) to compute the posterior probabilities for weighted average. However, equation 10 assumes that each frame can only be allocated to one and only one transform instead of using a weighted sum of posterior probabilities. This constraint comes from the way how CMLLR solves the equations which maintains the feature transformation is equivalent to model transformation. To remove such constraint, one can either uses quasi-Newton methods to optimize the transforms like the work in [6] or to solve a big system of linear equations. However, both methods are computationally expensive.

3.1. Context Transform for GDFT

As described, GDFT performs linear transformation on the feature vectors directly. In contrast, fMMI/MPE and RDLT can exploit the information available in the features within a context window, and also high dimensional posterior features. The linear transforms trained by fMMI/MPE and RDLT project the high dimensional features to the original feature space. The projection can be considered as some form of feature selection and it is optimized for some discriminative objective function. We propose an optimization algorithm for GDFT to perform a similar function, which allows GDFT to exploit the information available in different features.

Suppose we try to estimate a projection matrix P,

$$G(P) = \sum_i |Q_i(P) - C_i| + \frac{D}{2}||P - P^0||_F^2 , \quad (14)$$

where $Q_i(P) = \sum_t \sum_j \gamma_t^i(j)(Py_t - \mu_j)'\Sigma_j^{-1}(Py_t - \mu_j)$ is an auxiliary function to represent negative log likelihood; P^0 is the backoff projection. The projection matrix P projects the high dimensional feature y_t to the original feature space. y_t can be constructed using the original feature x_t. For example, $y_t = [x_{t-f}', \ldots, x_t', \ldots, x_{t+f}', 1]'$ where y_t is a supervector constructed by stacking the features within a context window of $\pm f$ frames. While there are many different ways to construct y_t, this paper focuses on the context features.

Similar to GBW and GDFT, we use Lagrange relaxation to solve equation 14, and we obtain the row-by-row update equation for P,

$$P_d = k_y^{(d)} G_y^{(d)-1} \quad (15)$$

where P_d is the d-th row of P, and,

$$G_y^{(d)} = \sum_i (\alpha_i - \beta_i) \sum_j \frac{1}{\sigma_{jd}^2} \sum_t \gamma_t^i(j) y_t y_t' + DI \quad (16)$$

$$k_y^{(d)} = \sum_i (\alpha_i - \beta_i) \sum_j \frac{\mu_{jd}}{\sigma_{jd}^2} \sum_t \gamma_t^i(j) y_t' + DP_d^0 \quad (17)$$

Similar to fMMI/MPE, the feature vectors are first transformed using the main transforms, W. Then, the features are stacked to form supervectors and we apply the projection as described in equation 15 to retrieve the final feature vectors in the feature space.

During training, the projection and the main transforms are jointly optimized. Although we can have multiple projections, we choose to have one projection transform and multiple main transforms like fMMI/MPE. For fMMI/MPE, only 10% of the training data is assigned to train the projection matrix. According to [3], it is to prevent the projection simply scales the transformed features. We adopt the same procedure for GDFT, which only 10% of the data is assigned to train the projection. In addition to solving the issues mentioned in [3], this also greatly speeds up the training process since for 90% of the data, GDFT operates on the low dimensional features, as computing G^d and k^d are much more efficient than computing G_y^d and k_y^d. One should note that this procedure does not benefit fMMI/MPE in terms of computation since fMMI/MPE uses gradient descent and the computation of the gradient must involve the high dimensional features.

4. EXPERIMENTAL SETUP

We conducted our experiments on two systems. Table 1 summarizes the configuration of these systems. Detailed system description of the Iraqi ASR is available in [7] and description of the MSA ASR system is available in [8]. For the experiments, 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 dev07/09 as dev sets, and eval09 and a three hours subset of dev10 as the unseen test sets.

	Iraqi ASR	MSA ASR
Train data	450 hr	1100 hr
System type	SA, 1-pass	SA, 3-pass
Vocab size	62K	737K
Adaptation	Incremental	Batch
# Gaussians	308K	867K
LM	3-gram	4-gram

Table 1. Description of the Iraqi and the MSA ASR systems.

We first compared regular discriminative training procedures with the single-pass discriminative training. For both regular and single-pass training, we use BMMI for model space discriminative training [9], and for feature space training, we use fMMI and our proposed GDFT. Both fMMI and GDFT have a context window of ± 7 frames and have 2048 main transforms. Performing fMMI followed by BMMI training is considered to be the state of the art for discriminative training.

Figure 1 shows that performing fMMI followed by BMMI (fMMI→BMMI) achieves 31.8% WER which improves the baseline ML model by 14.1% relative. If we replace fMMI with GDFT, we get 31.9% WER which is very similar to fMMI→BMMI. It is interesting to note that although fMMI outperforms GDFT alone, both training procedures give very similar performance after the model space training.

For single-pass training, we achieve 32.5% WER by using one EM iteration of GDFT and one EM iteration of BMMI using our recursive EBW algorithm with four M-steps (M=4). This performance is the same as the regular BMMI training without fMMI/GDFT (32.5%), but the regular BMMI training would need four passes on

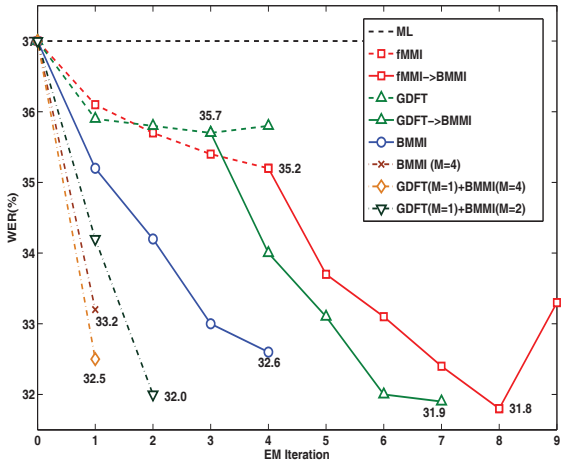

Fig. 1. Performance of different training procedures. This experiment is performed on the TransTac Jun08 open set using the Iraqi ASR system.

the train set instead of one. If we omit the GDFT for the single-pass training, the performance is 33.2% WER. In sum, our single-pass training achieves 86.5% of the total improvement available from discriminative training. If we release the single-pass constraint and allow two passes of the data, GDFT(M=1)+BMMI(M=2) gives 32.0% WER at the second EM iteration. This means it obtains 96.1% of the improvement available in the best training procedure (fMMI→BMMI). Table 2 summarizes the cost of each EM iteration for GDFT, fMMI and BMMI. We can see that discriminative training is very expensive but our proposed training procedure can drastically reduce the computation and yet, obtain most of the improvement from discriminative training.

fMMI	GDFT	BMMI
∼5 days	∼1 day	∼12 hours

Table 2. The time required for each EM iteration of fMMI, GDFT and BMMI on the 450-hr Iraqi train set. The benchmark was done on 20 CPU cores @ ∼2.66GHz. The costs reported here do not include the time for lattice generation. For single pass training using GDFT and BMMI, the time is similar to running GDFT alone.

Model/GDFT	M=1	M=2	M=3	M=4
ML	35.9%	35.6%	35.7%	35.7%
BMMI(M=4)	32.5%	33.0%	33.1%	33.0%

Table 3. The performance of single pass training with different combination of M-steps for GDFT and BMMI. The experiment is performed on TransTac Jun08 Open set.

Although we only performed one M-step for each EM iteration for GDFT in the experiment shown in figure 1, we tried the recursive update for GDFT as well. Table 3 shows the results of using different ways to combine GDFT and BMMI for single-pass training. When we use the ML model as the acoustic model, we observe GDFT can benefit from multiple M-steps. However, when we use the BMMI model (M=4) as the acoustic model, multiple M-steps

for GDFT would degrade the performance. This result is reasonable since when we train the model and the feature transforms jointly using single-pass training, BMMI is trained on the untransformed data, while GDFT assumes the acoustic model is the ML model. Hence, the mismatch becomes greater when we perform the recursive update. For single-pass training, we found that the best setup is one M-step for GDFT and four M-steps for BMMI.

	#iters	Jun08open	Nov08open
ML	-	37.0%	35.2%
BMMI	4	32.6%	30.6%
fMMI→BMMI	4+4	31.8%	30.0%
GDFT→BMMI	4+4	31.9%	30.0%
BMMI(M=4)	1	33.2%	31.3%
GDFT(M=1) +BMMI(M=4)	1	32.5%	31.0%
GDFT(M=1) +BMMI(M=2)	2	32.0%	30.5%

Table 4. The WER of the Iraqi ASR system on the Jun08 and the unseen Nov08 open sets.

	#iters	dev07	dev09	eval09	dev10
ML	-	13.7%	20.4%	15.1%	16.5%
BMMI	4	11.7%	18.6%	13.3%	14.6%
BMMI(M=4)	1	12.0%	18.6%	13.4%	14.7%
GDFT(M=1) +BMMI(M=4)	1	11.7%	18.5%	13.4%	14.6%

Table 5. The WER of the Vow 1100hrs 3-pass system on the GALE dev07/09/10 and eval09 test sets.

Table 4 and 5 show the performance of single-pass discriminative training on the Iraqi and the MSA speech recognition systems for different test sets. These tables also show the number of EM iterations used for different training procedures. The time required for each EM iteration for different algorithms is available in table 2. In sum, the results are consistent with the first experiment, which single-pass training using GDFT and fast EBW algorithm can achieve the performance of regular full BMMI training. If we allow two passes on the data, the performance of our proposed method is very close to the full fMMI and BMMI training.

5. CONCLUSION AND FUTURE WORK

We demonstrated how to combine our proposed recursive EBW algorithm and GDFT to achieve single-pass discriminative training. By processing the data only once, our proposed training procedure can achieve around 80% of the improvement available from full discriminative training. If we allow to process the data twice, we can obtain almost all of the improvement. This training procedure speeds up the process for building a speech recognition system, and enables us to build larger systems if data is available.

For the future work, we will explore online methods for discriminative training. Online methods for speech recognition are often applied on speaker adaptation for incremental improvement. For discriminative training, researchers may adopt online methods only when the optimization is performed using gradient based methods [10]. While the BW and EBW algorithms are designed to maximize an objective function over a batch of data and it may not help online training, we can try to exploit the regularization of EBW to perform incremental training. By combining the multiple updates strategy used by single-pass discriminative training, we may achieve more improvement by only processing the data once.

For discriminative training, we will continue to explore how to incorporate more information for GDFT. In this paper, we found that the performance of GDFT is comparable to fMMI, but we believe there is still room to improve GDFT. For example, the current implementation only uses context features, but it can also use posterior features like fMMI/MPE and RDLT. In sum, if GDFT can be improved, it will also help the performance of single-pass discriminative training.

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] R. Hsiao and T. Schultz, "Generalized Baum-Welch Algorithm and Its Implication to a New Extended Baum-Welch Algorithm," in *Proceedings of the INTERSPEECH*, 2011.

[2] R. Hsiao, F. Metze, and T. Schultz, "Improvements to Generalized Discriminative Feature Transformation for Speech Recognition," in *Proceedings of the INTERSPEECH*, 2010.

[3] D. Povey, "Improvements to fMPE for Discriminative Training of Features," in *Proceedings of the INTERSPEECH*, 2005, pp. 2977–2980.

[4] B. Zhang, S. Matsoukas, and R. Schwartz, "Recent Progress on the Discriminative Region-dependent Transform for Speech Feature Extraction," in *Proceedings of the INTERSPEECH*, 2006, pp. 1573–1576.

[5] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[6] S. S. Kozat, K. Visweswariah, and R. A. Gopinath, "Feature Adaptation based on Gaussian Posteriors," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2006, pp. 221–224.

[7] 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.

[8] 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.

[9] 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.

[10] C. Cheng, F. Sha, and L. K. Saul, "A Fast Online Algorithm for Large Margin Training of Continous-density Hidden Markov Models," in *Proceedings of the INTERSPEECH*, 2009.