Generalization of Extended Baum-Welch Parameter Estimation for Discriminative Training and Decoding

Dimitri Kanevsky[1], *Tara N. Sainath*[2], *Bhuvana Ramabhadran*[1], *David Nahamoo*[1]

[1]IBM T. J. Watson Research Center, Yorktown, NY 10598, U.S.A
[2]MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar St. Cambridge, MA 02139, U.S.A
{kanevsky, bhuvana, nahamoo}@ibm.us.com[1], tsainath@mit.edu[2]

Abstract

We demonstrate the generalizability of the Extended Baum-Welch (EBW) algorithm not only for HMM parameter estimation but for decoding as well. We show that there can exist a general function associated with the objective function under EBW that reduces to the well-known auxiliary function used in the Baum-Welch algorithm for maximum likelihood estimates. We generalize representation for the updates of model parameters by making use of a differentiable function (such as arithmetic or geometric mean) on the updated and current model parameters and describe their effect on the learning rate during HMM parameter estimation. Improvements on speech recognition tasks are also presented here.

1. Introduction

Efficient methods for estimating Hidden Markov Models (HMM) parameters are essential for solving a wide range of natural language processing tasks, such as part-of-speech tagging, word segmentation, optical character recognition, as well as acoustic modeling in speech recognition, just to name a few applications. Baum-Welch (BW) is a well-known efficient method for computing the maximum-likelihood estimates of the parameters of a HMM that is based on the existence of an auxiliary function that is a global lower bound of a likelihood function. Another efficient method is the EBW approach [1], [2] that is currently considered one of the most successful discriminative training techniques for estimating parameters of a HMM modeled with Gaussian mixtures. Initially EBW was introduced in [2] where it was suggested to modify a discriminative function by adding a "constant" everywhere in a probability domain function in such a way that the traditional BW technique could be applied.

In this paper, we explore the analogy between BW and EBW algorithms. We first show that there exists a function associated with an objective function (called the associated function) under EBW that is reduced to the auxiliary function used for maximum likelihood estimates. We also give a generalized representation for the updates of model parameters by making use of a differentiable function (such as arithmetic or geometric mean) on the updated (via an associated function) and current model parameters. EBW is an iterative algorithm with parameters that define the learning rate (or increase in the objective function) of parameters at every iteration thereby controlling the rate of convergence. These rules involve special EBW parameters that control the amount of change in an objective function (e.g. the Maximum Mutual Information Estimation (MMIE) objective) at each iteration of the algorithm.

Significant efforts in speech community has been devoted to learning what values of these control parameters lead to better estimation of parameters of Gaussian mixture in discriminative tasks. We provide a detailed theoretical analysis of the effect of these parameters that affect the learning rate during HMM parameter estimation while providing supporting experimental results on a large vocabulary speech recognition task.

The same generalized representation used earlier for updating model parameters during training, can be extended to be used in the decoding framework as a distance metric, instead of the traditionally used maximum-likelihood decoding. We provide experimental results on a phone classification task that establishes the benefits of using this representation. We also explore the use of traditional Maximum Likelihood Linear Regression (MLLR) and Maximum A Posteriori (MAP) updates within this representation. To summarize, this paper demonstrates the generalizability and versatility of the EBW algorithm both during training of HMM parameters and as a metric during decoding while providing significant improvements in performance on a speech recognition and phone classification task when compared to the performance of state-of-the-art systems in those domains.

The rest of the paper is structured as follows. In Section 2, we introduce the generalized representation and establish the analogy between EBW and BW while providing a family of EBW update rules. In Section 4, we present explicit formulas to measure the gradient steepness and explore its use in a decoding framework. In Section 5 we present application of EBW metrics to decoding tasks and extend them to use MAP and MLLR updates in the decoding metric. Experimental results are presented in Section 6. We conclude with a summary of the key messages in this work with suggestions for future work.

2. A New Family of EBW Update Rules

In this section we introduce a new family of update rules for parameter estimation of HMMs modeled with diagonal Gaussian mixtures. We also establish the relationship between EBW and BW algorithms with the use of an associated function.

Assume that data $y_i, i \in I = \{1, ..., n\}$ is drawn from a Gaussian mixture with each component of the mixture described by the parameters $\theta_j = (\mu_j, \sigma_j)$, where μ_j is the mean and σ_j is the variance od the jth component. Thus the probability of y_i given model θ_j is $z_{ij} = z_i(\theta_j) = \frac{1}{(2\pi)^{1/2}\sigma_j} e^{-(y_i-\mu_j)^2/2{\sigma_j}^2}$. Let $F(z) = F(\{z_{ij}\})$ be some objective function over $z = \{z_{ij}\}$, and let $c_{ij} = z_{ij}\frac{\delta}{\delta z_{ij}}F(z)$.

We will now define the following function that we will call

the function *associated* with F:

$$Q(\theta'_j, \theta_j) = \sum_i z_i(\theta_j) \frac{\delta F(\{z_i(\theta_j)\})}{\delta z_i(\theta_j)} \log z_i(\theta'_j),$$

Optimizing this function will lead to closed-form update rules (that are generally not obtainable if optimizing F directly). Let $\{\tilde{\theta}_j\}$ be solutions of $\frac{\delta Q(\theta'_j, \theta_j)}{\delta \theta'_j} = 0$.

Note that when the objective F is the log-likelihood function (e.g., standard MLE estimation in HMM, i.e. the Baum-Welch method), then Q coincides with the auxiliary function .

In this paper we introduce the following set of iterative rules for updating the current model parameters θ_j to their next values $\hat{\theta}_j$ (and a subsequent analysis of their properties):

$$\hat{\theta}_j(\alpha_j) = g_j(\alpha_j)\tilde{\theta}_j + (1 - g_j(\alpha_j))\theta_j + f_j(\alpha_j) \quad (1)$$

Here g_j is a differentiable function such that $g_j(0) = 0$ and $f_j(\alpha_j) = o(\alpha_j)$ where $o(\epsilon)$ implies that $\lim_{\epsilon \to 0} o(\epsilon)/\epsilon \to 0$).

In the case g_j is identical, one can show that there that the update rules in Equation (1) also include the following EBW rules as a special case.

$$\hat{\mu}_j = \mu_j(C) = \frac{\sum_{i \in I} c_{ij} y_i + C\mu_j}{\sum_{i \in I} c_{ij} + C} \quad (2)$$

$$\hat{\sigma}_j^2 = \sigma_j(C)^2 = \frac{\sum_{i \in I} c_{ij} y_i^2 + C(\mu_j^2 + \sigma_j{}^2)}{\sum_{i \in I} c_{ij} + C} - \hat{\mu}_j^2 \quad (3)$$

where $C = \frac{\sum_i c_{ij}}{\alpha_j}$.

Indeed, assuming $\sum_i c_{ij} \neq 0$ and $\alpha_j = \frac{\sum_i c_{ij}}{C}$ we have

$$\theta_j(C) = \hat{\theta}_j(\alpha_j) \quad (4)$$

Here $|f_j(\alpha_j)| < d/C^2$ for sufficiently large C and for some constant d. To show this inequality one needs to observe that $\tilde{\mu}_j = \frac{\sum_i c_{ij} y_i}{\sum_i c_{ij}}$ and $\hat{\mu}_j = \mu_j(0)\tilde{\alpha}_j + \mu_j(1 - \tilde{\alpha}_j)$ where $\tilde{\alpha}_j = \tilde{\alpha}_j(C) = \frac{\sum_i c_{ij}}{\sum_i c_{ij} + C}$ and $\mu_j(0)$ is defined as in (2) for $C = 0$. This implies statement (4) for the mean parameter). Statement (4) for the variance follows from the fact that $\tilde{\sigma}_j^2 = \frac{\sum_i c_{ij}(y_i - \mu_j)^2}{\sum_i c_{ij}}$ and from the linearized equations for variances in (18) in [2].

Example of applications of non-trivial g_j is given in the next section. Introduction of g_j in (1) allows to consider update rules to which techniques developed in [3] is applicable, and ,specifically, one can prove that iterative applications of rules 1 convergences to a local maximum of the objective function F.

2.1. Geometric Representation of EBW Update Rules

Let us connect models $\{\mu_j, \sigma_j\}$ and $\{\tilde{\mu}_j, \tilde{\sigma}_j\}$ with a curve segment (which generalizes the straight line segment used by [4]). Then the following cases for location of an updated model (at which F increases its value) can be considered: 1) If $\sum_i c_{ij} > 0$ then a step $\alpha_j > 0$ and $\{\hat{\mu}_j(\alpha_j), \hat{\sigma}_j(\alpha_j)\}$ lies on a segment that connects $\{\mu_j, \sigma_j^2\}$ and $\{\tilde{\mu}_j, \tilde{\sigma}_j^2\}$. 2) If $\sum_i c_{ij} < 0$ then a step $\alpha_j < 0$ and $\{\hat{\mu}_j(\alpha_j), \hat{\sigma}_j^2(\alpha_j)\}$ lies outside of the segment that connects $\{\mu_j, \sigma_j^2\}$ and $\{\tilde{\mu}_j, \tilde{\sigma}_j^2\}$. These cases correspond to cases in [4] where a sign of a step along a gradient was chosen depending on whether F has the minimum or the maximum at $\{\tilde{\mu}_j, \tilde{\sigma}_j^2\}$. The above process is illustrated in Fig. 1.

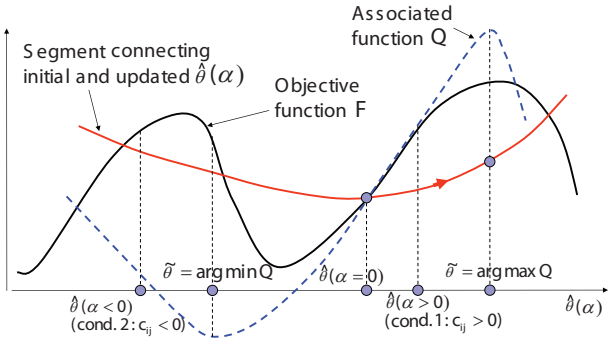

Figure 1: *Illustration of the new update rules*

2.2. Multiplicative Form of Update Rules

In keeping with the generalized representation of update rules, this section presents the update rules expressed in a multiplicative form (one form of the differentiable function g_j). In many applications (e.g. [4]) a weighted multiplicative form of update rules for variances was considered instead of a weighted sum (i.e. a geometric mean as opposed to an arithmetic mean as done in Equation 1). The corresponding update rules are: $\check{\theta}_j(\alpha'_j) = \tilde{\theta}_j^{\alpha'_j} * \theta_j^{(1-\alpha'_j)}$. It is easy to see that the multiplicative form of update rules is equivalent to the EBW family update rules given in Equation 1. Indeed, for every α' one can find such α that the following equality holds: $\check{\theta}_j = \tilde{\theta}_j^{\alpha'_j} * \theta_j^{(1-\alpha'_j)} = \hat{\theta}_j(\alpha_j) = \alpha_j\tilde{\theta}_j + (1-\alpha_j)\theta_j$ and can be solved as: $\alpha_j = \frac{\check{\theta}_j - \theta_j}{\tilde{\theta}_j - \theta_j}$ if $\theta_j \neq \tilde{\theta}_j$. Also one can easily see that if α'_j close to zero then $\check{\theta}_j$ is close to θ_j and therefore α_j also is close to zero. Therefore [3] implies that $F(\check{z}) \geq F(z)$ for sufficiently small α'_j (where $\check{z} = \{\check{z}_{ij}\}$ and $\check{z}_{ij} = \frac{1}{(2\pi)^{1/2}\tilde{\sigma}_j} e^{-(y_i - \tilde{\mu}_j)^2/2\tilde{\sigma}_j^2}$).

3. EBW Gradient Steepness Measurements

The same generalized representation discussed in Section 2 for updating model parameters during discriminative training, can be extended and used in the decoding framework as a distance metric to explain the quality of the model to fit the data. In this section, we derive the EBW gradient metric and explore different model update and learning rate methods within the gradient metric framework for decoding via HMMs.

3.1. Gradient Steepness Derivation

Using Equation 1 such that $\theta_j \to \hat{\theta}_j(C)$ (where C is inversely proportional to α) and $z_i(\theta_j) \to \hat{z}_i(\hat{\theta}_j)$, [3] derives the linearization formula between $F(\hat{z}_i(\hat{\theta}_j))$ and $F(z_i(\theta_j))$ for large C as:

$$F(\hat{z}_i(\hat{\theta}_j)) - F(z_i(\theta_j)) = T_i(\theta_j)/C + o(1/C) \quad (5)$$

Here T measures the gradient required to adapt the initial model, θ_j to data x_i. [3] also shows that T is always non-negative and only equals zero when $\hat{\theta}_j$ is a local maximum of $F(\hat{z}_i(\hat{\theta}_j))$. This guarantees that $F(z_i(\theta_j))$ increases per iteration and provides some theoretical justification for using gradient metrics T and $\left(F(\hat{z}_i(\hat{\theta}_j)) - F(z_i(\theta_j))\right) \times C$ as measures of quality of fitness of models to data.

A large value in T means the gradient is steep and $F(\hat{z}_i(\hat{\theta}_j))$ is much larger than $F(z_i(\theta_j))$. Thus the data is much better explained by the updated model $\hat{\theta}_j(C)$ compared to θ_j. However a small value in T indicates that the gradient is relatively flat and $F(\hat{z}_i(\hat{\theta}_j))$ is close to $F(z_i(\theta_j))$. Therefore, the initial model θ_j is a good fit for the data.

3.2. Model Update and Learning Rate Extensions

Given an HMM defined by a set of states $S = \{s_1, s_2 \dots s_N\}$ and observation sequence O, this HMM can be used in decoding tasks to find the most optimal state sequence through time $Q = \{q_1, q_2 \dots q_T\}$. We can define the probability that o_t came from from model θ_j in state s_j as $b_j(o_t)$. Typically, $b_j(o_t)$ is evaluated using standard maximum likelihood (i.e., $b_j(o_t) = P(o_t|q_t = s_j)$). $b_j(o_t)$ is evaluated across all frames and the most optimal state sequence is found via a dynamic programming Viterbi algorithm. Below, we discuss evaluating $b_j(o_t)$ in the gradient metric framework with new learning rate and model update methods.

3.2.1. EBW Gradient Metric

Instead of scoring $b_j(o_t)$ using likelihood, we can score it using the EBW gradient measurement given in Equation 5. Let us define objective function $F(z_t(\theta_j))$ to be the likelihood of observation o_t given state model θ_j as $F(z_t(\theta_j)) = P(o_t|q_t = s_j)$. Given o_t and initial model θ_j, we can re-estimate a new model $\hat{\theta}_j$ using the EBW update formulas given in Equations 2 and 3. Then, using Equation 5 and the objective function for $F(z_t(\theta_j))$, we compute the gradient metric score at frame o_t, normalized by the initial $F(z_t(\theta_j))$ as:

$$b_j(o_t) = \frac{\Big(F(\hat{z}_t(\hat{\theta}_j)) - F(z_t(\theta_j))\Big) \times C}{F(z_t(\theta_j))} \quad (6)$$

In [5], we compared scoring $b_k(o_t)$ using likelihood and the gradient metric given in Equation 6 for the recognition of Broad Phonetic Classes (BPC). Below, we discuss novel learning rate and model update methods in the EBW framework.

3.2.2. EBW Reduced Variance

In [5], the EBW re-estimate for $\hat{\theta}_j$ depended both on the adapted frame o_t and the initial model θ_j. The problem with this approach is that if $(F(\hat{z}_t(\hat{\theta}_j))$ has a lot of variance relative to $F(z_t(\theta_j))$ then the EBW distance criteria in Equation 6 may generate additional errors. Particularly because model re-estimation is done on a *per frame* basis, this problem is very likely. In order to solve this problem, we look to reduce the variation of the updated model $\hat{\theta}_j$ by weighting it by a factor $\beta < 1$, found during training, and shifting it by a constant N, also determined in training. This gives the updated model as:

$$\beta\hat{\theta}_j + (1 - \beta)N \quad (7)$$

We explore using this updated parameter estimate given in Equation 7 in our EBW gradient metric from Equation 6.

3.2.3. MAP Parameter Estimation

Instead of using the EBW parameter update formulas in Equations 2 and 3, we explore using the MAP [6] parameter estimates in our gradient steepness derivation. We express the MAP estimates by the following formula, given as an interpolation between initial and updated model parameters [6]:

$$\hat{\mu}_j = \alpha\frac{\sum_{i\in I} c_{ij}y_i}{\sum_{i\in I} c_{ij}} + (1 - \alpha)\mu_j \quad (8)$$

$$\hat{\sigma}_j^2 = \alpha\frac{\sum_{i\in I} c_{ij}(y_i - \hat{\mu}_j)^2}{\sum_{i\in I} c_{ij}} + (1-\alpha)((\mu_j - \hat{\mu}_j)^2 + \frac{2\beta_j}{C}) \quad (9)$$

where $\alpha = \frac{\sum_i c_{ij}}{\sum_i c_{ij}+C}$ and β_j is function of the initial variance as defined in [6]. The MAP and EBW mean updates are the same, while the MAP variance update scales β_j (and thus σ_j^2) by $1/C$ and thus gives less to σ_j^2 relative to EBW. We explore using these MAP updates in our EBW gradient metric.

3.2.4. MLLR

MLLR [7] is another common model adaptation technique. In this work, we explore expressing the updated model $\hat{\theta}_j = \{\hat{\mu}_j, \hat{\sigma}_j^2\}$ as a weighted combination of MLLR models and initial models. In other words

$$\hat{\theta}_j = \alpha\theta_{MLLR} + (1 - \alpha)\theta_j \quad (10)$$

Instead of estimating θ_{MLLR} on a per-frame basis, as in EBW and MAP, we explore estimating θ_{MLLR} on a per-utterance basis. However, the gradient metric using the MLLR update models is still scored on a per-frame basis.

4. Experiments

4.1. Discriminative Training for Broadcast News

Discriminative training experiments are performed on the speaker independent English broadcast news transcription system. Details on the experimental setup can be found in [1]. We report results on the rt03, dev04f and rt04 test sets as defined for the English portion of the EARS program.

We performed experiments testing various members in an EBW family for transformations where we varied f_j and ratio of α_j for means and variances in (1). Specifically, we investigated the following:

1. Linearized update of means: $\hat{\mu}_j = \mu_j(C_j) = \mu_j + \frac{\sum_{i\in I} c_{ij}(y_i - \mu_j)}{C_j}$

2. Ratio of control parameters: $D_j/C_j = 1.5$ where D_j is C_j for variance in (3)

3. Low value of control parameters: C_j for each Gaussian prototype is chosen to keep variance positive, e.g. starting from low $C_j = 1$ and multiplying C_j by 1.1 until variance becomes positive. It was observed that the best result occurs when all three conditions are combined, which also allows for faster training (See Section 5).

4.2. Recognition of Broad Phonetic Classes (BPC)

Our recognition experiments explore BPC recognition on the TIMIT corpus. The 61 TIMIT labels are mapped into 7 broad phonetic classes (BPC) as described in [5]. Each BPC is modeled as a three-state, left-to-right context-independent HMM. All models were trained on the standard NIST training set. We train the EBW gradient methods to find the appropriate model interpolation α and D weights using the dev set. We report recognition results on both the dev set and the full test set.

5. Results

5.1. Discriminative Training

Table 1 describes test set results in which we iterated over the discriminative training used in the baseline with modified EBW (3 above conditions) training for the 1st and 5th iterations. Columns labeled as *test name/baseline* (for example, *rt04/base*) contain results for baseline MMI training for 8 iterations (starting from a ML baseline). Each column labeled as *test name/mixed* represents two results. In a line *iteration 1* result of the application of a modified EBW method to the ML baseline is presented. These results show a single iteration in *rt04 mixed* achieved a WER of 18.9%, which is similar to the WER of *rt04 baseline* that was achieved at the 4th iteration. In other words, the first iteration of the modified EBW training allows to achieve decoding results that require 2-4 iterations of the baseline training. At *iteration 5*, the application of the modified EBW training to an output of a 4th iteration of the baseline are presented. These results are slightly better or the same that were obtained with 8 iterations of the baseline training.

	Test Set					
Iter	rt03 base	rt03 mixed	dev04f base	dev04f mixed	rt04 base	rt04 mixed
0	13.0		23.2		20.5	
1	12.6	**12.3**	22.4	**21.5**	19.9	**18.9**
2	**12.3**		21.8		19.5	
3	12.3		21.4		19.1	
4	12.2		21.3		18.8	
5	12.3	**12.0**	21.1	**21.1**	18.7	**18.3**
7	12.1		21.1		18.4	
8	**12.0**		**21.0**		**18.5**	

Table 1: English WER on test sets rt03, dev04f and rt04

These results show that the modified EBW in 1-2 iterations achieve the same decoding result that can be obtained in 3-4 iterations with the baseline method and therefore is much faster. We did not reproduce results with two or three subsequent iterations of modified EBW here since application of a subsequent iteration of modified EBW does not leads to significant improvement of the accuracy. This is because using low C and D in conditions 1-3 leads to overfitting. In order to avoid overfitting one needs to increase C,D significantly if they are run in consequent iterations with low C, D.

5.2. BPC Recognition

Table 5.2 shows the phonetic recognition error rates for the likelihood and gradient metrics, with the best performing metric indicated in bold. We investigate likelihood decoding using both initial baseline models and MLLR models, adapted per-utterance with the number of regression classes optimized on the dev set. We also explore decoding using the gradient steepness metric, where we explore model updates using EBW, EBW Reduced Variance, MAP and MLLR on a per-frame basis.

First, notice that the gradient metrics outperform both the baseline and MLLR likelihoods. This is because, as shown in Equation 6, the gradient metric captures the relative difference between the likelihood of the data given an initial model and a model estimated from the current data sequence being scored. Particular if the models are poor, the likelihood is only able to use one set of models, whether adapted or unadapted, and is unable to capture the change between initial and updated models.

In addition, when we can control the model learning rate, which is important when using 1 frame for model re-estimation, the Gradient-EBW Reduced Variance outperforms the other gradient metrics. Also, the importance of the variance term for model re-estimation can be observed in the slightly better performance for the Gradient-EBW metric relative to Gradient-MAP. Finally, we observe that using 1 frame to do model re-estimation with the Gradient-EBW metrics offers comparable performance to Gradient-MLLR where the models are re-estimated per-utterance.

Method	Dev	Test
Likelihood-Baseline Models	17.9	18.7
Likelihood-MLLR	18.1	19.0
Gradient-EBW	17.2	18.2
Gradient-EBW Reduced Variance	**16.9**	**18.2**
Gradient-MAP	17.3	18.3
Gradient-MLLR	17.2	18.3

Table 2: BPC Phonetic Error Rates on TIMIT

6. Conclusion and Future Work

In the paper we considered a family of EBW update rules that can be associated with weighted sums of updated and initial models. We demonstrate the generalizability of the Extended Baum-Welch (EBW) algorithm not only for HMM parameter estimation but for decoding as well. We show that there can exist a general function associated with the objective function under EBW that reduces to the well-known auxiliary function used in the Baum-Welch algorithm for maximum likelihood estimates. We generalized the representation for model parameter updates by making use of a differentiable function (such as arithmetic or geometric mean) on the updated and current model parameters and analyzed their effect on the learning rate. We provide experimental results on a BPC recognition that establishes the benefits of using this representation. We also explore the use of traditional Maximum Likelihood Linear Regression (MLLR) and Maximum A Posteriori (MAP) updates within this representation. We plan to continue to study EBW based training in which EBW control parameters are correlated to gradient steepness. We also plan to extend results of this paper for multivariate multidimensional Gaussian mixture densities.

7. References

[1] D. Povey and B.Kingsbury, "Evaluation of Proposed Modifications to MPE for Large Scale Discriminative Training," in *Proc. ICASSP*, 2007.

[2] P.S. Gopalakrishnan, D. Kanevsky, D. Nahamoo, and A. Nadas, "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems," *IEEE Trans. Information Theory*, vol. 37, no. 1, January 1991.

[3] D. Kanevsky, "Extended Baum Transformations For General Functions, II," Tech. Rep. RC23645(W0506-120), Human Language Technologies, IBM, 2005.

[4] C. Liu, P. Liu, H. Jiang F. Soong, and R. Wang, "Constrained Line Search Approach to General Discriminative HMM Training," in *ASRU*, 2007.

[5] T. N. Sainath, D. Kanevsky, and B. Ramabhadran, "Broad Phoentic Recognition in a Hidden Markov Model Framework Using Extended Baum-Welch Transformations," in *Proc. ASRU*, 2007.

[6] J.L. Gauvain and C.H. Lee, "Maximum a Posteriori Estimationg for Multivariaet Gaussian Observations of Markov Chains," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, 1994.

[7] C.J. Leggetter and P.C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, 1995.