RAPID ADAPTATION USING PENALIZED-LIKELIHOOD METHODS

Hakan Erdoğan, Yuqing Gao, and Michael Picheny

IBM TJ Watson Research Center
PO Box 218 Yorktown Heights, NY 10598
{erdogan,yuqing,picheny}@us.ibm.com

ABSTRACT

In this paper, we introduce new rapid adaptation techniques that extend and improve two successful methods previously introduced, cluster weighting (CW) and MAPLR. First, we introduce a new adaptation scheme called CWB which extends the cluster weighting adaptation method by including a bias term and a reference speaker model. CWB is shown to improve the adaptation performance as compared to CW. Second, we introduce an extension of cluster weighting that uses penalized-likelihood objective functions to stabilize the estimation and provide soft constraints. Third, we propose a variant of MAPLR adaptation that uses prior speaker information. Previously, prior distributions of transforms in MAPLR were obtained using the same adaptation data, speaker independent HMM means or by some heuristics. We propose to use the prior information of speaker variability to obtain the priors, by using CW or CWB weights. Penalized-likelihood or Bayesian theory serves as a tool to combine transformation based and prior speaker information based adaptation methods resulting in effective rapid adaptation techniques. The techniques are shown to outperform full, block diagonal and diagonal MLLR as well as some other recently proposed methods for rapid adaptation.

1. INTRODUCTION

A generic or speaker independent speech recognition system is suited for general purpose speech recognition. Speaker and environment adaptation is performed to adapt a generic system to a new speaker or environment, so that it performs better for the new input. Most current speaker adaptation techniques modify HMM state output distributions represented by Gaussian mixtures. Recent methods for speaker adaptation include MAP [1], MLLR [2], cluster weighting [3, 4], eigenvoices [5], and MAPLR [6, 7] among others.

The MAP method assumes a separate independent prior for each HMM Gaussian component, and a maximum *a posteriori* probability (MAP) objective function is maximized. Only the HMM components that were "seen" in the adaptation data can be adapted with this method. In contrast, MLLR method assumes that the new adapted component means are obtained through a linear transform of speaker independent means, coupling the parameters and enabling the "unseen" parameters to be adapted as well. The transforms are estimated using maximum likelihood. Another approach to adaptation is to use prior speaker information to estimate an adapted model. Speakers used in training the speech recognition system are utilized to obtain representative HMM models. Then, the adapted model means are represented as linear combinations of the means of these models and the interpolation weights are estimated by maximum likelihood. Representative HMMs can be obtained by clustering similar speakers together [8, 4] or by obtaining some eigenvectors that represent the most important information to construct a speaker dependent model [5]. In this paper, we use clustering to obtain reference models as in [8] and call the method "cluster weighting" (CW).

In this paper, we focus on the rapid adaptation problem, such as when the adaptation utterance is typically less than 5 seconds. Maximum Likelihood Linear Regression (MLLR) [2] method is better suited than MAP for this purpose. However, the amount of data to reliably estimate MLLR transform parameters is not enough, resulting in overtraining the parameters. MLLR can be suboptimal in this case. We believe, cluster weighting (CW) is more appropriate for rapid adaptation. Since the number of parameters to estimate are fewer, they can be estimated more reliably. On the other hand, MAPLR adaptation regularizes the linear regression transform estimation by using Bayesian theory. We propose a different approach to MAPLR prior estimation in this paper.

In the following sections, we first give a background on MLLR and CW adaptation approaches. We introduce an improved version of CW called CWB. We also propose use of the penalized-likelihood method to improve cluster weighting approaches. In the second part of the paper, we introduce a new way to obtain prior distribution parameters for MAPLR adaptation. We propose to use the cluster weights to obtain prior distributions for linear regression transformations in MAPLR. We present encouraging rapid adaptation results for new proposed algorithms.

2. BACKGROUND

Most current speech recognition systems use a continuous density HMM model with each state output distribution represented by diagonal covariance Gaussian mixtures. Assume, we have M unshared mixture components in our system. The output distribution corresponding to a single mixture component m is given by:

$$N(o;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)=\frac{1}{(2\pi)^{n/2}|\boldsymbol{\Sigma}_m|}\exp\{-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_m)^T\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_m)\},$$

where $\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$ are mean and covariance of the component m and n is the dimension of the feature vector. The output distribution for an HMM state is given by $\sum_{m\in\mathcal{S}}\omega_m N(o;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m)$ where ω_m are mixture weights and $\mathcal{S}$ represents the set of components belonging to a state.

2.1. Maximum Likelihood Linear Regression (MLLR)

In MLLR adaptation, the Gaussian distribution means are updated with an affine transformation:

$$\boldsymbol{\mu}_m^{\mathrm{MLLR}}=\boldsymbol{A}\boldsymbol{\mu}_m^{\mathrm{SI}}+\boldsymbol{b}=\boldsymbol{W}\boldsymbol{\xi}_m^{\mathrm{SI}},\quad\forall m\in\mathcal{R},$$

where the transform matrix $\boldsymbol{A}$ and bias vector $\boldsymbol{b}$ are shared among components $m \in \mathcal{R}$, $\boldsymbol{\xi}_m = [\boldsymbol{\mu}_m^T 1]^T$ is the extended mean vector and $\boldsymbol{W} = [\boldsymbol{A}\ \boldsymbol{b}]$ is the extended transform matrix. For simplicity, we ignore the dependence of the transformation parameters on the regression class $\mathcal{R}$ in our notation. There are usually multiple transforms $\{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_R\}$, each of which are applied to the HMM components in their respective regression classes $\{\mathcal{R}_1, \ldots, \mathcal{R}_R\}$.

2.2. Cluster Weighting (CW)

To perform CW adaptation, the training data is clustered into K groups each of which consist of "similar" speakers. A separate model for each cluster is obtained. In this paper, we assume each cluster model is obtained by using MLLR adaptation on the speaker independent means. However, it is possible to obtain cluster models by other adaptation methods (such as MAP) or even by direct EM training on the cluster data if enough data for a cluster exists. We denote cluster dependent model means by $\boldsymbol{\mu}_m^k$ for $k = 1, \ldots, K$. Since the cluster means were obtained by MLLR, we have:

$$\boldsymbol{\mu}_m^k = \boldsymbol{A}^k \boldsymbol{\mu}_m^{\mathrm{SI}} + \boldsymbol{b}^k = \boldsymbol{W}^k \boldsymbol{\xi}_m^{\mathrm{SI}}, \quad \forall m \in \mathcal{R}', \tag{1}$$

where $\boldsymbol{A}^k$ and $\boldsymbol{b}^k$ represent the MLLR transforms for cluster k and regression class $\mathcal{R}'$.

The adapted model is represented by:

$$\boldsymbol{\mu}_m^{\mathrm{CW}} = \sum_{k=1}^{K} \lambda_k \boldsymbol{\mu}_m^k = \boldsymbol{M}_m \boldsymbol{\lambda}, \quad \forall m \in \mathcal{R}'', \tag{2}$$

where $\boldsymbol{M}_m = [\boldsymbol{\mu}_m^1, \ldots, \boldsymbol{\mu}_m^K]$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^T$ is the vector of weights. The weights are shared among Gaussian components $m \in \mathcal{R}''$ similar to MLLR.

2.3. Estimation

Let $\boldsymbol{O} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_T\}$ represent acoustic feature vectors for the adaptation utterance. Adaptation can be performed by maximizing the likelihood of the adapted HMM model. In this paper, we only focus on adaptation methods that modify the means of the output distributions. The EM method is used to maximize (increase) the likelihood for computational simplicity. The EM auxiliary function can be written by:

$$Q(\theta) = K_1 - K_2 \sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_m(t) \log N(\boldsymbol{o}_t; \boldsymbol{\mu}_m^{\mathrm{SA}}(\theta), \boldsymbol{\Sigma}_m), \tag{3}$$

where θ are the parameters that define the adapted means and are to be estimated (*e.g.* $\boldsymbol{W}$ for MLLR and $\boldsymbol{\lambda}$ for CW), $\gamma_m(t)$ is the *a posteriori* probability of occupying component m given all the adaptation data $\boldsymbol{O}$, the current output distributions $\{\omega_m, \boldsymbol{\mu}_m^{\mathrm{SI}}, \boldsymbol{\Sigma}_m\}$ and transition probabilities[1]. $\gamma_m(t)$ is found by the forward-backward algorithm. K_1 and K_2 are constants for the purpose of this optimization problem.

When the auxiliary function Q is maximized, we obtain a closed form solution for both MLLR and CW adaptation when diagonal covariance Gaussian mixture components are used. For MLLR case, we get a closed form solution for rows of $\boldsymbol{W}$ matrix given by [2]:

$$\boldsymbol{W}_{i\cdot} = \boldsymbol{G}_i^{-1} \boldsymbol{z}_i,$$

where $\boldsymbol{W}_{i\cdot}$ is the ith row of $\boldsymbol{W}$ and

$$\boldsymbol{G}_i = \sum_{m \in \mathcal{R}} \frac{1}{\sigma_{m_i}^2} c_m \boldsymbol{\xi}_m \boldsymbol{\xi}_m^T, \quad \boldsymbol{z}_i = \sum_{m \in \mathcal{R}} \frac{1}{\sigma_{m_i}^2} d_{mi} \boldsymbol{\xi}_m,$$

where we define the "counts" as $c_m = \sum_{t=1}^{T} \gamma_m(t)$ and the "counts for mean" as $\boldsymbol{d}_m = \sum_{t=1}^{T} \gamma_m(t) \boldsymbol{o}_t$, and $\mathcal{R}$ is the regression class or the set of components that share a common transform.

For CW adaptation, the weight vector $\boldsymbol{\lambda}$ can be estimated by maximum likelihood as [4]:

$$\boldsymbol{\lambda} = \boldsymbol{G}_w^{-1} \boldsymbol{k}_w, \tag{4}$$

where

$$\boldsymbol{G}_w = \sum_{m \in \mathcal{R}''} c_m \boldsymbol{M}_m \boldsymbol{\Sigma}_m^{-1} \boldsymbol{M}_m^T, \quad \boldsymbol{k}_w = \sum_{m \in \mathcal{R}''} \boldsymbol{M}_m^T \boldsymbol{\Sigma}_m^{-1} \boldsymbol{d}_m. \tag{5}$$

Note that, the weights do not have to sum up to one and they might be negative as well as positive since this is an unconstrained estimation.

To determine the regression classes $\mathcal{R}$, a hierarchical bottom up clustering tree is used. Each leaf of the tree is a component or a set of components in the HMM. Nodes are merged using acoustic similarity measures. Then, we specify a threshold t_c for the total counts at each node (*i.e.* $\sum_m c_m$'s) and the node that first exceeds that threshold determines a regression class (*e.g.* see [9]).

3. CLUSTER WEIGHTING + BIAS (CWB)

The form of (2) is restrictive in that it assumes the adapted means to be in the span of the cluster means. However, if the channel conditions for the adaptation data are much different than the training data, this assumption may not hold. Since filtering in the time domain corresponds to addition in the cepstral domain, a bias term is added to the feature vectors if MFCC features (or linearly processed MFCC features) are used. This effect can be corrected by incorporating a bias term in the CW formulation.

A second shortcoming of the CW formulation is that it does not contain the speaker independent means. There might be cases such that the SI means represent the new speaker better than all the cluster means. So, we center our new CW adaptation method on a reference model $\boldsymbol{\mu}^{\mathrm{REF}}$ which represents a "best guess" model which is believed to perform best on the new speaker. So, we introduce the following cluster weighting + bias (CWB) adaptation method:

$$\boldsymbol{\mu}_m^{\mathrm{CWB}} = \boldsymbol{\mu}_m^{\mathrm{REF}} + \sum_{k=1}^{K} \lambda_k (\boldsymbol{\mu}_m^k - \boldsymbol{\mu}_m^{\mathrm{REF}}) + \boldsymbol{b}. \tag{6}$$

Usually, we choose $\mu_m^{\mathrm{REF}} = \mu_m^{\mathrm{SI}}$.

CWB estimation can be carried out in a similar fashion as CW estimation. CWB is still an interpolation of various vectors to obtain an adapted model. Equation (6) can be rewritten as:

$$\boldsymbol{\mu}_m^{\mathrm{CWB}} = \boldsymbol{\mu}_m^{\mathrm{REF}} + \boldsymbol{M}'_m \boldsymbol{\lambda}',$$

where $\boldsymbol{M}'_m = [\boldsymbol{\mu}_m^1 - \boldsymbol{\mu}_m^{\mathrm{REF}}, \ldots, \boldsymbol{\mu}_m^K - \boldsymbol{\mu}_m^{\mathrm{REF}}, \boldsymbol{I}]$ and $\boldsymbol{\lambda}' = [\boldsymbol{\lambda}^T, \boldsymbol{b}^T]^T$. Then, the equations (4) and (5) could be used to estimate $\boldsymbol{\lambda}'$ (*i.e.* $\boldsymbol{\lambda}$ and $\boldsymbol{b}$) except that $\boldsymbol{d}_m$ should be replaced by $\boldsymbol{d}_m - c_m \boldsymbol{\mu}_m^{\mathrm{REF}}$ in equation (5).

[1] It is possible to iterate the EM algorithm in this case by recalculating $\gamma_m(t)$ with adapted models and recomputing the next iteration adapted models using new values.

4. PENALIZED-LIKELIHOOD CLUSTER WEIGHTING

Since equation (3) is an unconstrained objective function, the weights $\boldsymbol{\lambda}$ in CW and CWB adaptation can take any value. However, it might be a good idea to put some "soft" constraints on their values for stability of the algorithm when only very little adaptation data is available. Penalized-likelihood provides a framework for incorporating such constraints. We add penalty terms to the log-likelihood function to penalize deviations from our soft constraints. This is equivalent to Bayesian estimation (MAP) when the penalty functions are seen as the logarithm of the prior distribution on the parameters. However, it is more convenient for us to adopt the penalized-likelihood view. In penalized-likelihood, instead of maximizing the log-likelihood $L(\theta)$, we minimize $-L(\theta) + R(\theta)$ where R is the penalty function. For CW adaptation, we minimize $-Q(\boldsymbol{\lambda}) + \sum_i \beta_i R_i(\boldsymbol{\lambda})$ where β_i represent how much each penalty function is weighted. We define the following penalty function for CW:

$$2R(\boldsymbol{\lambda}) = \beta_1(\sum_k \lambda_k - 1)^2 + \beta_2 \sum_k \lambda_k^2.$$

We make sure the penalty functions are quadratic, so that the optimization is easier. R_1 encourages the weights sum up to one and R_2 discourages negative weights if we assume R_1 is satisfied. Any of the β's could be set to zero to ignore that penalty function. Additional penalty functions can be added if more information is available for the new speaker. Reasonable values of β's could be determined by experimentation.

This penalized-likelihood objective is optimized directly with a formula similar to (4) :

$$\boldsymbol{\lambda} = (\boldsymbol{G}_w + \beta_1 + \beta_2 \boldsymbol{I})^{-1} (\boldsymbol{k}_w + \beta_1).$$

For CWB adaptation, the following penalty function makes sense:

$$R(\boldsymbol{\lambda}') = \beta_1 \sum_k \lambda_k^2 + \beta_2 \sum_i \boldsymbol{b}_i^2.$$

In this case, setting β_1 and β_2 to large numbers will encourage the weights and the bias term to be close to zero, in which case the adaptation will favor the reference model (μ^{REF}) and will not deviate a lot from it which might be a good idea if we have very short adaptation data. To remove the constraints on the weights and bias terms, the beta values could be set to zero, in which case the adapted model will deviate from the reference model and will be equivalent to the maximum likelihood CWB solution.

5. MAPLR WITH PRIORS THROUGH CLUSTER WEIGHTING

MAPLR is an extension of the MLLR method of adaptation which was introduced recently [6, 7]. In MAPLR formulation, a prior distribution on the transformation parameters $\boldsymbol{W}$ is assumed. The estimation is done by maximizing a maximum *a posteriori* (MAP) objective function instead of the usual maximum likelihood. It is shown that a closed form solution is possible when the distribution is chosen from the family of elliptically symmetric matrix variate distributions [6]. In practice, diagonal covariance Gaussian priors are used and the priors are estimated from data. MAPLR with matrix variate Gaussian priors is equivalent to penalized-likelihood when the penalty term is a sum of weighted least squares functions for the rows of the transformation matrix.

$$\phi(\boldsymbol{W}) = -Q(\boldsymbol{W}) + \beta \sum_i (\boldsymbol{W}_{i\cdot} - \hat{\boldsymbol{m}}_i)^T \boldsymbol{S}_i^{-1} (\boldsymbol{W}_{i\cdot} - \hat{\boldsymbol{m}}_i),$$

where $\boldsymbol{W}_{i\cdot}$ denotes ith row of the matrix $\boldsymbol{W}$, $\hat{\boldsymbol{m}}_i$ is the prior mean and $\boldsymbol{S}_i$ is the prior covariance for the ith row of $\boldsymbol{W}$. Here β is a hyperparameter that determines how much weight is to be given to the penalty function (or the log-prior). It is equivalent to weighting the prior (co)variance matrix $\boldsymbol{S}_i$ by $1/\beta$ in Bayesian method.

In previous studies [6, 7], only priors obtained from the adaptation data itself or speaker independent model means were used. Structural MAPLR [9] makes use of the regression tree as an additional information source. In this paper, we propose to find the priors (or the penalty term parameters) using the cluster weights from CW or CWB adaptation. We make use of the prior training speaker information *unlike* all the other MAPLR methods. In some way, this is a hybrid between cluster weighting and transform based adaptation. Since the clusters were obtained by MLLR transformations as in equation (1), we can write the CW adapted means as follows:

$$\mu_m^{\mathrm{CW}} = \sum_k \lambda_k \boldsymbol{A}^k \boldsymbol{\mu}_m^{\mathrm{SI}} + \sum_k \lambda_k \boldsymbol{b}^k = \sum_k \lambda_k \boldsymbol{W}^k \boldsymbol{\xi}_m^{\mathrm{SI}}.$$

This is equivalent to applying the affine transformation $\sum_k \lambda_k \boldsymbol{W}^k$ directly to the SI means. We choose this transform to be the prior mean for MAPLR, *i.e.*

$$\hat{\boldsymbol{m}}_i = \sum_k \lambda_k \boldsymbol{W}_{i\cdot}^k.$$

The prior variance terms $\boldsymbol{S}_i$'s can be obtained by many ways. Using the weights to obtain prior variance is not a good idea, since weights might be negative, but absolute value or square of the weights could be used. We use the following simple formula to find the prior diagonal covariance matrix for row i:

$$(\boldsymbol{S}_i)_{jj} = 1/K \sum_k (\boldsymbol{W}_{ij}^k - \hat{\boldsymbol{m}}_{ij})^2. \tag{7}$$

When we use CWB adaptation to find the priors, we naturally use the following equation for the prior mean transformation when $\boldsymbol{\mu}^{\mathrm{REF}} = \boldsymbol{\mu}^{\mathrm{SI}}$:

$$\hat{\boldsymbol{m}} = \left[\boldsymbol{I} + \sum_k \lambda_k (\boldsymbol{A}^k - \boldsymbol{I}),\ (\sum_k \lambda_k \boldsymbol{b}^k) + \boldsymbol{b} \right].$$

The variance computation remains the same as (7).

The estimation formula for MAPLR is given by:

$$\boldsymbol{W}_{i\cdot} = (\boldsymbol{G}_i + \beta \boldsymbol{S}_i^{-1})^{-1} (\boldsymbol{z}_i + \beta \boldsymbol{S}_i^{-1} \hat{\boldsymbol{m}}_i).$$

Note that the statistic $\boldsymbol{G}_i$ is proportional to c_m, the adaptation counts. So, when we have a lot of counts, $\boldsymbol{G}_i$ dominates the denominator, however in a low count case, the prior term dominates. The level of domination depends on the value of β which can be determined with experimentation for a given system.

In practice, the transform or weight sharing could be different for CW and MAPLR adaptation methods. Since we ignored regression classes in our notation, this is not apparent. However, it is easy to remedy this situation by using the hierarchical tree nature. We usually have MAPLR count threshold (t_c) to be much lower than CW or CWB threshold[2], so we use the cluster weights from parent nodes in MAPLR regression classes.

[2]The reason for this is that the MAPLR transform will favor the prior transform if we have low count anyway, so there is no reason to have a high threshold which will result in a coarse set of transforms.

6. RESULTS

We have tested the methods introduced in this paper on IBM name dialer test data. Each utterance is very short about 2-5 seconds which qualifies for rapid adaptation tests. The test data has 5190 name dialer calls. We compared our methods with other popular rapid adaptation techniques. The results are shown in Table 1.

The starting system is a generic telephony system trained with 600K sentences of telephony data (8KHz) from different domains. The system uses an LDA matrix applied to 9 spliced frames of 13 MFCC components each to reduce the dimension to 39. The decoding parameters were optimized for the telephony data. The vocabulary size was about 8000 words. We applied "massive adaptation" to this generic system to obtain a massive adapted system. We used a pool of name dialing data as adaptation data (about 10K calls) and adapted the generic system to the names domain. This reduced the word error rate considerably from 11.21% to 9.53%. We consider this adapted system to be the baseline system for us. For use in CW and CWB adaptation, we performed speaker clustering of the training speakers. For each speaker, we computed a simple one Gaussian per context-independent phone state model and used k-means on the means to cluster speakers into groups. We generated an 8 cluster system.

Starting with the baseline system, we applied *unsupervised* adaptation to each utterance separately. We first decoded the utterance using the baseline system, then used the decoded script to compute statistics for adaptation. After adapting the baseline model with various methods, we decoded the same call using the adapted models. Hence, the adaptation and the test data were the same in our tests. We used maximum likelihood estimation for CW and CWB adaptation. Penalized-likelihood with different β values were used for MAPLR experiments. The regression thresholds t_c were obtained by heuristics and past experience.

Method	t_c	WER	UER
Generic telephony system	-	11.21	12.37
Massive adapted (MLLR+MAP)	800	9.53	10.52
MLLR (full)	800	9.39	10.37
MLLR (diagonal)	200	9.21	10.21
MLLR (2x2 block diagonal)	300	8.83	9.90
DLLR (λ=0.1)	200	8.96	9.83
CW ($K = 8$)	100	9.12	10.10
MAPLR, CW prior ($\beta = 50$)	5	8.33	9.27
MAPLR, CW prior ($\beta = 100$)	5	8.42	9.44
CWB ($K = 8$)	200	8.79	9.81
MAPLR, CWB prior ($\beta = 1$)	5	8.80	9.87
MAPLR, CWB prior ($\beta = 50$)	5	8.59	9.56
MAPLR, CWB prior ($\beta = 100$)	5	8.65	9.61
MAPLR, CWB prior ($\beta = 200$)	5	8.73	9.79

Table 1: Utterance adaptation results for name dialer test data. Word error rate (WER) and utterance error rate (UER) in percentages are shown. Baseline is the massive adapted system. t_c is the count threshold to determine regression classes $\mathcal{R}$. Triple line separates old methods and newly proposed methods.

The results show that block diagonal MLLR performed the best among old rapid adaptation methods. Discounted likelihood linear regression (DLLR) [10] also performed better than full and diagonal MLLR. CWB outperforms CW method, but the best result is obtained with MAPLR with CW priors and $\beta = 50$ which achieves a 12% reduction in the WER reducing it to 8.33% from 9.53%.

7. CONCLUSION

We introduced new penalized-likelihood methods for rapid adaptation of speech recognizers. Our methods extend and improve previously introduced cluster weighting and MAPLR techniques for adaptation. We introduced CWB method that adds bias and reference speaker terms to the CW method. We propose to use penalized-likelihood cluster weight estimation for stability. We introduced a new way to obtain priors for MAPLR adaptation using CW or CWB weights. The new methods outperform other techniques in an unsupervised rapid adaptation scenario for a name recognition task.

The adaptation methods we tried were performed on the HMM model component means. In real applications of telephony speech recognition, it is very expensive to update the models for each utterance. There are a class of feature space transformations that achieve similar effect as the model space transformations [11]. Feature space transforms are cheaper in terms of computation and storage. In our future work, we plan to study feature space transform counterparts of our methods.

REFERENCES

[1] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Tr. Speech and Audio Proc.*, vol. 2, no. 2, pp. 291–99, April 1994.

[2] 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, pp. 171–85, 1995.

[3] M. J. F. Gales, "Transformation smoothing for speaker and environmental adaptation," in *European Conference on Speech Communication and Technology*, 1997.

[4] M. J. F. Gales, "Cluster adaptive training for speech recognition," in *International Conference on Spoken Language Processing*, 1998.

[5] R. Kuhn, P. Nguyen, J. C. Junqua, R. Boman, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Fast speaker adaptation using a priori knowledge," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, pp. 749–52, 1999.

[6] W. Chou, "Maximum a posteriori linear regression with elliptically symmetric matrix variate priors," in *European Conference on Speech Communication and Technology*, 1999.

[7] C. Chesta, O. Siohan, and C.-H. Lee, "Maximum a posteriori linear regression for hidden Markov model adaptation," in *European Conference on Speech Communication and Technology*, 1999.

[8] Y. Gao, M. Padmanabhan, and M. Picheny, "Speaker adaptation based on pre-clustering training speakers," in *European Conference on Speech Communication and Technology*, 1997.

[9] O. Siohan, T. A. Myrvoll, and C.-H. Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," in *Automatic Speech Recognition and Understanding Workshop*, 2000.

[10] W. Byrne and A. Gunawardana, "Discounted likelihood linear regression for rapid adaptation," in *European Conference on Speech Communication and Technology*, 1999.

[11] M. J. F. Gales, "Linear transform interpolation with HMMs," Technical report, Cambridge University Engineering Department, 2000.