A BASIS METHOD FOR ROBUST ESTIMATION OF CONSTRAINED MLLR

Daniel Povey and Kaisheng Yao

Microsoft, One Microsoft Way, Redmond, WA
{dpovey,kaisheny}@microsoft.com

ABSTRACT

Constrained Maximum Likelihood Linear Regression (CMLLR) is a widely used speaker adaptation technique in which an affine transform of the features is estimated for each speaker. However, when the amount of speech data available is very small (e.g. a few seconds), it can be difficult to get sufficiently accurate estimates of the transform parameters. In this paper we describe a method of estimating CMLLR robustly from less data. We do this by representing the CMLLR transform matrix as a weighted sum over basis matrices, where the basis is constructed in such a way that the most important variation is concentrated in the leading coefficients. Depending on the amount of data available, we can choose to estimate a smaller or larger number of coefficients.

***Index Terms*—** Speech Recognition, Speaker Adaptation, MLLR

1. INTRODUCTION

Constrained Maximum Likelihood Linear Regression (CMLLR) [1, 2] is a popular form of speaker adaptation, in which an affine transform is applied to the speech features:

$$\mathbf{x} \rightarrow \mathbf{A}^{(s)}\mathbf{x} + \mathbf{b}^{(s)}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^D$ is the feature vector, and $\mathbf{A}^{(s)}$ and $\mathbf{b}^{(s)}$ are transformation parameters specific to speaker s. We will write this here in the more convenient form

$$\mathbf{x} \rightarrow \mathbf{W}^{(s)}\mathbf{x}^{+}, \tag{2}$$

where $\mathbf{x}^+ = \left[\mathbf{x}^T \ , \ 1\right]^T$, and $\mathbf{W}^{(s)} = \left[\mathbf{A}^{(s)} \ ; \ \mathbf{b}^{(s)}\right]$. CMLLR was originally described as a model-space transform; because it can also be represented as a feature-space transform, it is sometimes known as feature-space MLLR (fMLLR).

CMLLR is typically estimated by Maximum Likelihood; an EM algorithm described in [2] is commonly used to estimate it. When the amount of adaptation data available is very small (e.g. less than about five seconds), the parameters cannot be robustly estimated and CMLLR does not lead to improvements in Word Error Rate (WER). Various methods have been proposed to improve CMLLR estimation for limited adaptation data. These include the use of block-diagonal and diagonal forms of the matrix $\mathbf{A}$ [2], the use of Bayesian priors ("fMAPLR") [3, 4], and representing $\mathbf{W}$ in a smaller dimension using a basis [5, 6]. In [6] it was found that it is important to train such a basis using a Maximum Likelihood criterion rather than the Principal Components Analysis (PCA) scheme previously used. In Section 2 we discuss this prior work in more detail.

In Section 3 we describe the key ideas behind our approach. We represent $\mathbf{W}^{(s)}$ using a sum over a set of basis matrices. The general idea is the same as [5] but we have solved a number of problems with the original approach to make it efficient and to ensure that we do not degrade results when a lot of adaptation data is available. Space constraints do not permit a detailed description of our algorithm, which we have presented in [7] along with more detailed experiments.

We present our experimental results in Section 4. We compare our method with standard CMLLR, diagonal and block-diagonal CMLLR, and fMAPLR. Our experiments show a clear advantage of our technique over these baselines. We conclude in Section 5.

2. PRIOR WORK ON ROBUST CMLLR ESTIMATION

Various methods have been proposed for robust adaptation on small amounts of data. In [2] it was mentioned that diagonal or block-diagonal structures for $\mathbf{A}$ can be used to reduce the number of parameters to estimate. Because of their simplicity such methods are frequently used, and we use them as baselines here.

Bayesian techniques were investigated in [3] and [4], both under the name fMAPLR. The basic idea is to use the Maximum A Posteriori rule to choose the parameter, given a suitable prior, i.e. to maximize:

$$p(\mathbf{W}|\mathcal{X}) \propto p(\mathcal{X}|\mathbf{W})p(\mathbf{W}), \tag{3}$$

where $\mathcal{X}$ is the speech data, $p(\mathcal{X}|\mathbf{W})$ is the data likelihood, and $p(\mathbf{W})$ is the prior likelihood. The two papers both used Gaussian priors over $p(\mathbf{W})$ (viewing the matrix as the vector of its concatenated rows) but they used different ways of compactly representing the prior covariance which is a large matrix of dimension $D(D{+}1) \times D(D{+}1)$ where D is the feature dimension. In [4] a factor-analyzed form was used (i.e. the covariance matrix was a diagonal matrix plus the outer product of a rectangular matrix), and in [3] a diagonal matrix was used. In both cases the Maximum Likelihood estimates of the matrices $\mathbf{W}^{(s)}$ for a set of speakers were used as training data for the prior parameters (a simple "empirical Bayes" approach). The version of fMAPLR we used as a baseline here is a slight generalization of [3], in which we give the covariance of the prior a block-diagonal structure (one block for each row of $\mathbf{W}$) and also introduced a scaling factor on the log-prior term, which we tuned to optimize WER. We trained the prior only on speakers with a relatively large amount of adaptation data, as we found this worked best.

A basis representation of the CMLLR matrix was described in [5]. The idea is to represent $\mathbf{W}$ as a sum over basis matrices:

$$\mathbf{W}^{(s)} = \sum_{n=1}^{N} d_n^{(s)} \mathbf{W}_n, \tag{4}$$

where N is some basis size decided in advance with $1 \leq N < D(D{+}1)$ (e.g. $N{=}200$), $\mathbf{W}_n$ are the basis matrices, and $d_n^{(s)}$ are speaker-specific coefficients. This improved WER for small amounts of adaptation data, but the only baseline reported was conventional CMLLR, and the technique ultimately degraded performance as the amount of adaptation data became larger (due to the fixed basis size). In [6] the same idea was pursued further, and it was found that for best performance it was important to train the matrices $\mathbf{W}_n$ in a Maximum Likelihood fashion. The method described there was not very practical because the EM algorithm used to train the basis matrices was extremely slow.

Basis reprepresentations have also been proposed for conventional Maximum Likelihood Linear Regression (MLLR), e.g. Eigen-MLLR [8]. The disadvantage of such approaches is that they are difficult to make very efficient, since they require the model's means to be transformed for each new speaker. This will typically dominate the computation time in cases where the amount of adaptation data is very small. Another adaptation method suited to fast adaptation is Eigenvoices [9], but methods of that type are not very practical due to the very large number of parameters to be learned in training time.

3. KEY IDEAS OF OUR APPROACH

The basis representation we use is very similar to (4), except with an offset term and (more importantly) a basis size that varies per speaker:

$$\mathbf{W}^{(s)} = \mathbf{W}_0 + \sum_{n=1}^{N(s)} d_n^{(s)} \mathbf{W}_n, \tag{5}$$

where $0 \leq N(s) \leq D(D{+}1)$ and $\mathbf{W}_0 = [\mathbf{I} \ ; \ \mathbf{0}]$. In our work we just set $N(s)$ to be proportional to the the amount of adaptation data (but not exceeding $D(D{+}1)$ which is the number of parameters in $\mathbf{W}$). Note that while this is a model selection problem, we have not compared against standard model selection methods such as the Bayesian Information Criterion (BIC), or the Aikake Information Criterion (AIC). This is because, in our experience, for these kinds of problems, tunable selection criteria such as BIC do not perform very differently from simple count-based heuristics. Non-tunable criteria like the AIC are problematic for speech tasks due to the extent of model incorrectness. Our approach to setting $N(s)$ was chosen for simplicity and speed.

Probably the key aspects of our work that distinguish it from [5, 6] are the use of a varying number of basis elements, and our approach to computing the basis matrices $\mathbf{W}_n$. This approach approximates Maximum Likelihood but is still efficient and is applicable when the basis size is to be decided in test time. The preconditioning we use to accomplish this has the useful side effect that it speeds up the algorithms we use to learn the parameters $d_n^{(s)}$ in test time. Some of the ideas used here are derived from prior work described in [10], which describes an efficient method of updating the CMLLR transformation for a differently structured GMM-based system with full covariances, called a Subspace Gaussian Mixture Model (SGMM).

Below we discuss the ideas behind various aspects of our algorithm. In Section 3.1 we discuss the preconditioning; in Section 3.2 we describe how we compute the basis matrices $\mathbf{W}_n$; in Section 3.3 we describe how, in test time, we decide the value of $N(s)$ and compute the coefficients $d_n^{(s)}$.

3.1. Preconditioning

In many of our computations it is easiest to think of $\mathbf{W}$ as a vector rather than a matrix, so we define

$$\mathbf{w} = \mathrm{vec}\,(\mathbf{W}^T), \tag{6}$$

where the vec operator stacks the columns, so with the transpose, $\mathbf{w}$ is a row stack of $\mathbf{W}$; the transpose is useful later on. We will implicitly make use of (6), by making it apply to pairs $\mathbf{w}$ and $\mathbf{W}$ whenever they share the same subscripts, superscripts and other modifiers. Consider a second-order Taylor expansion of the auxiliary function, taken around $\mathbf{w} = \mathbf{w}_0$. We write $\Delta\mathbf{w}$ for $(\mathbf{w}{-}\mathbf{w}_0)$. The approximation is written in the following form:

$$\mathcal{Q}^{(s)}(\mathbf{w}) \simeq K + (\Delta\mathbf{w})^T \mathbf{p}^{(s)} - \tfrac{1}{2}(\Delta\mathbf{w})^T \mathbf{H}^{(s)} (\Delta\mathbf{w}), \tag{7}$$

where the quantities $\mathbf{p}^{(s)}$ and $\mathbf{H}^{(s)}$ may be computed from the CMLLR statistics [7]. The idea is to precondition via a change of variables, such that when written in the new variable, $\mathbf{H}^{(s)}$ has good condition number (i.e. it is close to the unit matrix times a constant). This is quite straightforward to do. First we define

$$\mathbf{H} = \frac{1}{\sum_s \beta^{(s)}} \sum_s \mathbf{H}^{(s)}, \tag{8}$$

where $\beta^{(s)}$ is the data count for speaker s, so $\mathbf{H}$ is the average value of the $\mathbf{H}^{(s)}$ term (normalized by the number of frames). Note that this is a $D(D{+}1) \times D(D{+}1)$ matrix. We do the Cholesky decomposition

$$\mathbf{H} = \mathbf{C}\mathbf{C}^T, \tag{9}$$

with $\mathbf{C}$ a lower triangular matrix. We then perform a change of variables by defining

$$\hat{\mathbf{w}} = \mathbf{C}^T \mathbf{w}. \tag{10}$$

Thus we can rewrite (7) as

$$\hat{\mathcal{Q}}^{(s)}(\hat{\mathbf{w}}) = (\Delta\hat{\mathbf{w}})^T \hat{\mathbf{p}}^{(s)} - \tfrac{1}{2}(\Delta\hat{\mathbf{w}})^T \hat{\mathbf{H}}^{(s)} (\Delta\hat{\mathbf{w}}), \tag{11}$$

via appropriate definitions of $\hat{\mathbf{p}}^{(s)}$ and $\hat{\mathbf{H}}^{(s)}$. We can show that in the transformed space, the Hessian averages to the unit matrix (i.e. $\hat{\mathbf{H}} = \mathbf{I}$).

3.2. Basis computation

The basis computation also relies on the Taylor approximation of (7). We additionally make the assumption that $\mathbf{H}^{(s)} \simeq \beta^{(s)}\mathbf{H}$ (equivalent to $\hat{\mathbf{H}}^{(s)} \simeq \beta^{(s)}\mathbf{I}$). This is reasonable as long as all the speakers are sufficiently similar. This assumption is necessary in order to reduce the problem to a PCA problem, which is tractable. These approximations may seem quite crude, but the key is that they make the basis computation fast and practical. In [6] more exact and expensive methods were considered, but they were too slow to be practical.

Under these assumptions it is easy to compute a Maximum Likelihood solution for the basis matrices. The way we formulate the problem is to ask for a set $\{\mathbf{W}_n, \, 1 \leq n \leq D(D{+}1)\}$, such that whatever basis size $1 \leq N < D(D{+}1)$ we choose, the training data likelihood (subject to our assumptions and approximations) is maximized. Let us consider the problem in its vector form (i.e. in terms of $\mathbf{w}_n$). In order to ensure good condition of the auxiliary function when written in terms of the coefficients d_n, we insist that the transformed form of the vectors (i.e. $\hat{\mathbf{w}}_n$) form an orthonormal set. We will consider some fixed but arbitrary basis size $1 \leq N \leq D(D{+}1)$, and write $\hat{\mathbf{w}}$ as a sum over basis elements, i.e.:

$$\hat{\mathbf{w}} = \hat{\mathbf{w}}_0 + \sum_{n=1}^{N} d_n \hat{\mathbf{w}}_n. \tag{12}$$

Here, $\hat{\mathbf{w}}_0$ is the transformed version of the identity feature-mapping $\mathbf{W}_0$. What we are doing is limiting $\Delta\hat{\mathbf{w}} \equiv \hat{\mathbf{w}} - \hat{\mathbf{w}}_0$ to the subspace spanned by the vectors $\hat{\mathbf{w}}_n$. We will now describe how we compute the basis elements $\hat{\mathbf{w}}_n$ in such a way that they approximately maximize the objective function for all basis sizes simultaneously.

We first write down the auxiliary function in $\hat{\mathbf{w}}$ without yet applying the subspace constraint of (12). Rewriting (11) using $\hat{\mathbf{H}}^{(s)} \simeq \beta^{(s)}\mathbf{I}$,

$$\hat{\mathcal{Q}}^{(s)}(\hat{\mathbf{w}}) \simeq (\Delta\hat{\mathbf{w}})^T\hat{\mathbf{p}}^{(s)} - \tfrac{1}{2}\beta^{(s)}(\Delta\hat{\mathbf{w}})^T(\Delta\hat{\mathbf{w}}). \qquad (13)$$

It is easy to see that this is maximized by $\Delta\hat{\mathbf{w}}^{(s)} = 1/\beta^{(s)}\hat{\mathbf{p}}^{(s)}$, and that the corresponding auxiliary function value is $1/(2\beta^{(s)})\ \hat{\mathbf{p}}^{(s)\ T}\ \hat{\mathbf{p}}^{(s)}$. Defining

$$\hat{\mathbf{M}} = \sum_s \frac{1}{\beta^{(s)}}\hat{\mathbf{p}}^{(s)}\ \hat{\mathbf{p}}^{(s)\ T}, \qquad (14)$$

we may write the total auxiliary function, summed over all speakers, as $\mathrm{tr}\,(\hat{\mathbf{M}})/2$. Suppose we write $\mathbf{X}_N = [\hat{\mathbf{w}}_1 \ldots \hat{\mathbf{w}}_N]$, with the $\hat{\mathbf{w}}_n$ orthonormal, to represent a basis of size N. It is not hard to show that when limiting $\hat{\mathbf{w}}$ to the form (12), the corresponding total auxiliary function value is $\mathrm{tr}\,(\mathbf{X}_N^T\hat{\mathbf{M}}\mathbf{X}_N)/2$, and that the basis $\mathbf{X}_N$ that maximizes this can be obtained by doing an eigenvalue decomposition on $\hat{\mathbf{M}}$ and letting $\hat{\mathbf{w}}_n$ be the n'th eigenvector of $\hat{\mathbf{M}}$ (ordered from largest to smallest eigenvalue), so that whatever basis size N we choose, $\mathbf{X}_N$ always contains the top N eigenvectors.

Thus, with the help of the preconditioning and some additional approximations, we have reduced the difficult Maximum Likelihood problem considered in [6] to a much easier PCA problem. After computing the vectors $\hat{\mathbf{w}}_n$ we can reverse the co-ordinate changes and un-stack the matrix columns to obtain the set $\{\mathbf{W}_n, 1 \leq n \leq D(D{+}1)\}$.

3.3. Test time computation

In test time, the optimization problem is as follows: we are given the speaker-specific statistics $\mathbf{K}^{(s)}$, $\mathbf{G}_i^{(s)}$ and $\beta^{(s)}$ (c.f. [2]), the basis $\{\mathbf{W}_n, 1 \leq n \leq D(D{+}1)\}$ and a basis size $0 \leq N(s) \leq D(D{+}1)$ and we need to estimate the speaker-specific coefficients $d_n^{(s)}$ of (5).

Our update is iterative. On each iteration $1 \leq q \leq Q$ (where e.g. $Q = 10$), we compute the gradient of the auxiliary function w.r.t. the coefficients $\{d_n^{(s)}, 1 \leq n \leq N(s)\}$ and select a search direction. In the basic version of our method the search direction is just the gradient direction; we also tried a modification based on the conjugate gradient method. Then we do a line search in that direction; our line search is iterative and based on Newton's method (in one dimension). Both our basic method and the conjugate gradient modification converge very fast and are guaranteed not to decrease the auxiliary function; results here are with the basic version.

The number of coefficients we use was determined by the formula $N(s) = \min(\lfloor \eta\beta^{(s)} \rfloor, D(D{+}1))$, where η is a constant set by hand (e.g. $\eta = 0.2$) that controls how many parameters we add for each new frame of data.

4. EXPERIMENTAL RESULTS

4.1. System descriptions

We used two systems to evaluate our techniqe: an "Interactive Voice Response" (IVR) system, trained for telephone-based voice interface applications, for which the test data mostly contained short utterances, and an "Enhanced Voice Mail" (EVM) system, trained for transcribing voice mails, where the test data was mostly longer utterances. We further split the test sets into different duration bins, in order to see how our technique performs with different amounts of test data.

Both systems use cross-word triphone context dependency, with standard phone context clustering and three-state left-to-right HMMs. The features are 13-dimensional MFCCs with Δ, $\Delta\Delta$ and $\Delta\Delta\Delta$ (i.e. deltas, accelerations and third derivatives), reduced in dimension with HLDA. We train with cepstral mean normalization; in test time this is applied in an on-line fashion. We do not use VTLN, but rely instead on gender-dependent models. The gender-independent models are used in the first pass of decoding to obtain the supervision hypothesis and the corresponding phone-level alignments.

The IVR system was trained on 7500 hours of speech, mostly voice search data recorded over the telephone but also read speech. The average length of training utterances is 5.3 seconds. The feature dimension after HLDA is 36. Each of the three (GI, male and female) models has 9116 clustered states with on average 46 Gaussians per state. The models were trained with Minimum Classification Error (MCE). The test set contains three subsets consisting of digits, city-state pairs and stock names, totaling 20 hours of speech, with 21K words and an average utterance length of 3.4 seconds.

The EVM system is as the IVR system but with 33 dimensional feature vectors and trained with MMIE. The number of clustered states is 10144, with on average 47 Gaussians per state. The training data consists of 1700 hours of read speech with an average utterance length of 5.3 seconds, plus 130 hours of voicemail recordings with an average utterance length of 35.3 seconds (this data is scaled up by a factor of 20 in training). Our test set contains five different sources of voicemail test data, totaling 507 utterances with 4 hours of speech in total. The average length of the test utterances is 28.4 seconds.

We apply the CMLLR estimation (in both the baselines and our technique) in a segment-wise online fashion. That is, we divide the utterances up into segments and the CMLLR estimation for each segment only sees the statistics for that segment and preceding segments. For the statistics accumulation, the CMLLR transform estimated from the previous segments is used for within-phone alignment (the phone-level segmentation is fixed by the first pass decoder). The segmenter is tuned to give about three segments per utterance.

	IVR				EVM			
Mean Duration (s)	2.2	3.8	5.4	7.7	11.6	28.1	51	101
#Words	29K	22K	30K	12K	7K	11K	12K	6.5K
#Utterances	12.6K	3.7K	3.5K	1.4K	210	138	81	23
Length (h)	7.8	3.9	5.2	2.9	0.68	1.08	1.14	0.64
%WER (Unadapted)	5.64	1.63	0.65	0.98	32.8	31.4	33.3	35.8

Table 1. Utterance duration bins and baseline WERs

In order to see how our technique performs on different utterance lengths, we broke up the IVR and EVM test sets into four subsets each, corresponding to different duration bins. Table 1 describes these bins, along with the baseline (unadapted) WERs in each bin. There is a very large variation in WERs because different bins are dominated by different types of test data. In particular, the longer IVR bins are dominated by digits which have very low error rates.

4.2. Baselines

The baseline adaptation strategies we compare against are "full CMLLR", which refers to CMLLR estimated in the standard way, "block-diagonal CMLLR", in which the matrix $\mathbf{A}$ is block diagonal with three equal-sized blocks, and "diagonal CMLLR" where

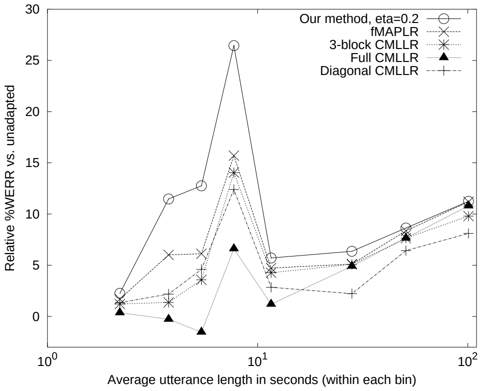

Fig. 1. WER improvement from adaptation vs. utterance duration

$\mathbf{A}$ is diagonal. Since we use HLDA features there is no special meaning to the three blocks; however, it seems to work in practice. We also compare with fMAPLR [3], for which we use a separate full-covariance Gaussian for each row; we extend it by putting a scale on the log prior term and tuning the scale. For the IVR system our fMAPLR is applied to block diagonal CMLLR with 3 blocks, and for the EVM system we use a full matrix; this choice was made to optimize WER. The prior used for fMAPLR in the 3-block case is a full-covariance Gaussian that models the non-zero part of each row, which is a vector of dimension $D/3 + 1$. We have limited our baselines that those that we might conceivably use in practice, which ruled out [5, 6] because they degrade results when the amount of adaptation data is large (and they are also quite complex).

4.3. Results

Figure 1 shows the relative WER improvement of various CMLLR adaptation strategies, compared with using no CMLLR adaptation, for the duration bins described in Table 1. It can be seen that our method performs best, followed by fMAPLR, followed by standard CMLLR; depending on the utterance length, either full, block-diagonal or diagonal CMLLR was best. The points on the left half of the graph are from IVR, and those on the right half are from EVM. The various bins of data are at very different absolute WERs (see Table 1, last row), and the relative improvements tend to be higher when the absolute WER is low. This explains the wide variations seen in Fig. 1. The line with full CMLLR (triangles) substantiates our claim that CMLLR does not give improvements below about five seconds of data.

Until about 20 seconds of adaptation data, our technique gives a substantial improvement over fMAPLR and the other baselines, and after that the differences are small. In addition, for less than about 3 seconds of data (first bin), none of the adaptation methods give very much improvement. The improvement that our method gives versus CMLLR is greatest between about 5 and 15 seconds of speech. For less than 20 seconds of speech our technique gives more than twice the improvement of fMAPLR, taking CMLLR as the baseline. For 20 seconds or more, the differences are small. Taking into account the online manner in which the CMLLR transforms are estimated, these statements should be modified when generalizing to systems in which the CMLLR is estimated after seeing the whole utterance. Our estimate is that the greatest improvement from our method will be between about 3 and 10 seconds of adaptation data.

The test time computation needed for our method is faster than the conventional approach to CMLLR estimation; see [7] for more details. Since CMLLR estimation is in any case very fast, this is not a significant practical advantage.

We ran matched-pairs significance tests comparing our method to CMLLR, and to fMAPLR, in each of the eight duration bins. Against CMLLR, our method gives significant improvements (at the 90% level) in all but the longest-duration bin. Against fMAPLR, our method only gives significant improvements for the third, fourth and sixth bins, but if we do the significance test for IVR as a whole, and EVM as a whole, it is significant in both cases.

5. CONCLUSIONS

We have described a practical algorithm for estimating Constrained MLLR transforms robustly by training a set of basis matrices and, in test time, restricting the estimated matrix to a leading subset of the basis matrices. Stated very crudely, our method reduces the amount of adaptation data that is needed to obtain a substantial improvement, from about 10 seconds to about 3 seconds. We have shown that it gives significant improvements versus the next best baseline we tested, fMAPLR, and is faster than conventional CMLLR. In [7] we have described our algorithm in a way which is intended to make it easy to reproduce our results.

6. REFERENCES

[1] V. Digalakis, D. Rtischev, and L. Neumeyer, “Fast speaker adaptation using constrained estimation of Gaussian mixtures,” *IEEE Trans. on Speech and Audio Processing*, pp. 357–366, 1995.

[2] M.J.F Gales, “Maximum Likelihood Linear Transformations for HMM-based Speech Recognition,” *Computer Speech and Language*, vol. 12, pp. 75–98, 1997.

[3] X. Lei, J. Hamaker, and X. He, “Robust Feature Space Adaptation for Telephony Speech Recognition,” in *Proc. ICSLP*, 2006.

[4] J. Huang, E. Marcheret, and K. Visweswariah, “Rapid Feature Space Speaker Adaptation for Multi-Stream HMM-Based Audio-Visual Speech Recognition,” *IEEE Int’l Conf on Multimedia and Expo*, pp. 338–341, 2005.

[5] K. Visweswariah, V. Goel, and R. Gopinath, “Structuring Linear Transforms for Adaptation Using Training Time Information,” in *Proc. ICASSP*, 2002.

[6] K. Visweswariah, V. Goel, and R. Gopinath, “Maximum Likelihood Training Of Bases For Rapid Adaptation,” in *Proc. ICSLP*, 2002.

[7] D. Povey and K. Yao, “A Basis Representation of Constrained MLLR Transforms for Robust Adaptation,” *Computer Speech and Language (accepted)*, 2011.

[8] K. T. Chen, W. W. Liau, H. M. Wang, and L. S. Lee, “Fast Speaker Adaptation Using Eigenspace-based Maximum Likelihood Linear Regression,” in *Proc. ICSLP*, 2000.

[9] R. Kuhn, F. Perronnin, P. Nguyen, and L. Rigazzio, “Very Fast Adaptation with a Compact Context-Dependent Eigenvoice Model,” in *Proc. ICASSP*, 2001.

[10] A. Ghoshal, D. Povey, et al., “A Novel Estimation of Feature-space MLLR for Full Covariance Models,” 2010, Proc. ICASSP.