A NOVEL ESTIMATION OF FEATURE-SPACE MLLR FOR FULL-COVARIANCE MODELS

Arnab Ghoshal[1], *Daniel Povey*[2],
Mohit Agarwal[3], *Pinar Akyazi*[4], *Lukáš Burget*[5], *Kai Feng*[6], *Ondřej Glembek*[5], *Nagendra Goel*[7],
Martin Karafiát[5], *Ariya Rastrow*[8], *Richard C. Rose*[9], *Petr Schwarz*[5], *Samuel Thomas*[8]

[1] Saarland University, Germany, arnab.ghoshal@lsv.uni-saarland.de;
[2] Microsoft Research, USA, dpovey@microsoft.com; [3] IIIT Allahabad, India;
[4] Boğaziçi University, Turkey; [5] Brno University of Technology, Czech Republic; [6] Hong Kong UST;
[7] Go-Vivace Inc., USA; [8] Johns Hopkins University, USA; [9] McGill University, Canada

ABSTRACT

In this paper we present a novel approach for estimating feature-space maximum likelihood linear regression (fMLLR) transforms for full-covariance Gaussian models by directly maximizing the likelihood function by repeated line search in the direction of the gradient. We do this in a pre-transformed parameter space such that an approximation to the expected Hessian is proportional to the unit matrix. The proposed algorithm is as efficient or more efficient than standard approaches, and is more flexible because it can naturally be combined with sets of basis transforms and with full covariance and subspace precision and mean (SPAM) models.

Index Terms— Speech recognition, Speaker adaptation, Hidden Markov models, Optimization methods, Linear algebra

1. INTRODUCTION

Feature-space MLLR (fMLLR), also known as Constrained MLLR, is a popular adaptation technique used in automatic speech recognition to reduce the mismatch between the models and the acoustic data from a particular speaker [1]. The method uses an affine transform of the feature vectors,

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

where the transform parameters $\mathbf{A}^{(s)}$ and $\mathbf{b}^{(s)}$ are estimated to maximize the likelihood of the transformed speaker-specific data under the acoustic model. In this paper we use the following compact notation, $\mathbf{x}^{(s)} = \mathbf{W}^{(s)}\mathbf{x}^{+}$, where $\mathbf{W}^{(s)} = [\mathbf{A}^{(s)}, \mathbf{b}^{(s)}]$, and $\mathbf{x}^{+} = [\mathbf{x}^{\mathrm{T}}, 1]^{\mathrm{T}}$ denotes a vector $\mathbf{x}$ extended with a 1.

For each speaker transform $\mathbf{W}^{(s)}$, we need to estimate d^2+d parameters, where d is the dimensionality of the feature vectors. When small amounts of per-speaker adaptation data are present, these estimates may not be reliable. In such cases we can express $\mathbf{W}^{(s)}$ as a linear combination of a set of *basis* transforms $\mathbf{W}_b$ ($1 \leq b \leq B$),

$$\mathbf{W}^{(s)} = \mathbf{W}_0 + \textstyle\sum_{b=1}^{B} \alpha_b^{(s)} \mathbf{W}_b. \tag{2}$$

This work was conducted at the Johns Hopkins University Summer Workshop which was (partially) supported by National Science Foundation Grant Number IIS-0833652, with supplemental funding from Google Research, DARPA's GALE program and the Johns Hopkins University Human Language Technology Center of Excellence. BUT researchers were partially supported by Czech MPO project No. FR-TI1/034. Thanks to CLSP staff and faculty, to Tomas Kašpárek for system support, to Patrick Nguyen for introducing the participants, to Mark Gales for advice and HTK help, and to Jan Černocký for proofreading and useful comments.

This is similar to the formulation presented in [2]. The bases $\mathbf{W}_b$ are estimated from the entire training set, and essentially define a *subspace* for the fMLLR transforms. For each speaker we only need to estimate the coefficients α_b, where typically $B \ll d(d+1)$.

When the underlying probability model has a diagonal covariance structure, the fMLLR transform can be estimated by iteratively updating the matrix rows [1]; a similar row-by-row update has also been proposed for the full covariance case [3]. Such updates are hard to combine with the subspace formulation of Equation (2).

We propose to estimate the transformation matrices $\mathbf{W}^{(s)}$ by repeated line search in the gradient direction, in pre-transformed parameter space where the Hessian is proportional to the unit matrix. Our experiments are with a new kind of GMM-based system, a "subspace Gaussian mixture model" (SGMM) system [4]. For the purposes of fMLLR estimation, the only important fact is that this system uses a relatively small number (less than 1000) of shared full-covariance matrices. However, our approach is also applicable to diagonal and other types of systems (e.g. diagonal-covariance, full-covariance, SPAM [5]).

2. SUBSPACE GMM ACOUSTIC MODEL

The simplest form of SGMM can be expressed as follows:

$$p(\mathbf{x}|j) = \textstyle\sum_{i=1}^{I} w_{ji}\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_i) \tag{3}$$

$$\boldsymbol{\mu}_{ji} = \mathbf{M}_i \mathbf{v}_j \tag{4}$$

$$w_{ji} = \frac{\exp(\mathbf{w}_i^{\mathrm{T}}\mathbf{v}_j)}{\sum_{i'=1}^{I}\exp(\mathbf{w}_{i'}{}^{\mathrm{T}}\mathbf{v}_j)}, \tag{5}$$

where $\mathbf{x} \in \Re^D$ is the feature, j is the speech state, and $\mathbf{v}_j \in \Re^S$ is the "state vector" with $S \simeq D$ being the subspace dimension. The model in each state is a GMM with I Gaussians whose covariances $\boldsymbol{\Sigma}_i$ are shared between states. The means $\boldsymbol{\mu}_{ji}$ and mixture weights w_{ji} are not parameters of the model. Instead, they are derived from $\mathbf{v}_j \in \Re^S$, via globally shared parameters $\mathbf{M}_i$ and $\mathbf{w}_i$. Refer to [4, 6] for details about the model, parameter estimation, etc.

3. ESTIMATION OF FMLLR TRANSFORM

The auxiliary function for estimating the fMLLR transform $\mathbf{W}$ by maximum likelihood, obtained by the standard approach using Jensen's inequality, is as follows:

$$\mathcal{Q}(\mathbf{W}) = \sum_{t\in\mathcal{T}(s),j,i} \gamma_{ji}(t)[\log|\det\mathbf{A}| - \ldots$$

$$\frac{1}{2}(\mathbf{W}\mathbf{x}^+(t) - \boldsymbol{\mu}_{ji})^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\mathbf{W}\mathbf{x}^+(t) - \boldsymbol{\mu}_{ji})] \quad (6)$$

$$\mathcal{Q}(\mathbf{W}) = C + \beta \log|\det \mathbf{A}| + \mathrm{tr}\,(\mathbf{W}\mathbf{K}^{\mathrm{T}}) - \ldots \frac{1}{2}\sum_i \mathrm{tr}\,\left(\mathbf{W}^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}\mathbf{W}\mathbf{G}_i\right) \quad (7)$$

where C is the sum of the terms without $\mathbf{W}$, $\mathcal{T}(s)$ is the set of frames for speaker s, and β, $\mathbf{K}$, $\mathbf{G}_i$ are the sufficient statistics that need to be accumulated:

$$\beta = \sum_{t\in\mathcal{T}(s),j,i} \gamma_{ji}(t) \quad (8)$$

$$\mathbf{K} = \sum_{t\in\mathcal{T}(s),j,i} \gamma_{ji}(t)\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_{ji}\mathbf{x}^+(t)^{\mathrm{T}} \quad (9)$$

$$\mathbf{G}_i = \sum_{t\in\mathcal{T}(s),j,i} \gamma_{ji}(t)\mathbf{x}^+(t)\mathbf{x}^+(t)^{\mathrm{T}} \quad \text{(type 1)} \quad (10)$$

$$\mathbf{G}_k = \sum_{t\in\mathcal{T}(s),j,i} \gamma_{ji}(t)p_{jik}\mathbf{x}^+(t)\mathbf{x}^+(t)^{\mathrm{T}} \quad \text{(type 2).} \quad (11)$$

The "type 1" formulation is efficient for SGMM where we have a relatively small number of full covariance matrices. The "type 2" formulation is a more general approach that covers SPAM systems [5] (in which $\boldsymbol{\Sigma}_{ji}$ are represented as a weighted combination of basis inverse variances $\mathbf{A}_k$ for $1 \le k \le K$, with $\boldsymbol{\Sigma}_{ji}^{-1} = \sum_k p_{jik}\mathbf{A}_k$), as well as diagonal and general full covariance systems. This notation is however not optimal for diagonal and full-covariance systems.

Defining $\mathbf{P} \in \Re^{d\times(d+1)}$ as the derivative of $\mathcal{Q}(\mathbf{W})$ with respect to $\mathbf{W}$, we have:

$$\mathbf{P} \equiv \frac{\partial\mathcal{Q}(\mathbf{W})}{\partial\mathbf{W}} = \beta\left[\mathbf{A}^{-\mathrm{T}};\, \mathbf{0}\right] + \mathbf{K} - \mathbf{G}, \quad (12)$$

$$\mathbf{G} = \sum_i \boldsymbol{\Sigma}_i^{-1}\mathbf{W}\mathbf{G}_i \quad \text{(type 1)} \quad (13)$$

$$\mathbf{G} = \sum_k \mathbf{A}_k\mathbf{W}\mathbf{G}_k \quad \text{(type 2)} \quad (14)$$

This gradient is computed on each iteration of an iterative update process.

4. HESSIAN COMPUTATION

In general we would expect the computation of the matrix of second derivatives (the Hessian) of the likelihood function $\mathcal{Q}(\mathbf{W})$ to be difficult. However, with appropriate pre-scaling and assumptions this $d(d+1)$ by $d(d+1)$ matrix has a simple structure. The general process is that we pre-transform so that the means have diagonal variance and zero mean and the average variance is unity; we make the simplifying assumption that all variances are unity, and then compute the *expected* Hessian for statistics generated from the model around $\mathbf{W} = [\mathbf{I}\,;\,\mathbf{0}]$ (i.e. the default value). We then use its inverse Cholesky factor as a pre-conditioning transform. We never have to explicitly construct a $d(d+1)$ by $d(d+1)$ matrix.

4.1. Pre-transform

To compute the transform matrix $\mathbf{W}_{\mathrm{pre}} = [\mathbf{A}_{\mathrm{pre}}\ \mathbf{b}_{\mathrm{pre}}]$ we start by calculating the average within-class and between-class covariances, and the average mean, as in the LDA computation:

$$\boldsymbol{\Sigma}_W = \sum_i \gamma_i\boldsymbol{\Sigma}_i \quad (15)$$

$$\boldsymbol{\mu} = \frac{1}{\sum_{j,i}\gamma_{j,i}}\sum_{j,i}\gamma_{ji}\boldsymbol{\mu}_{ji} \quad (16)$$

$$\boldsymbol{\Sigma}_B = \left(\frac{1}{\sum_{j,i}\gamma_{j,i}}\sum_{j,i}\gamma_{ji}\boldsymbol{\mu}_{ji}\boldsymbol{\mu}_{ji}^{\mathrm{T}}\right) - \boldsymbol{\mu}\boldsymbol{\mu}^{\mathrm{T}}, \quad (17)$$

where $\gamma_{ji} = \sum_t \gamma_{ji}(t)$, and $\gamma_i = \sum_j \gamma_{ji}$. We first do the Cholesky decomposition $\boldsymbol{\Sigma}_W = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, compute $\mathbf{B} = \mathbf{L}^{-1}\boldsymbol{\Sigma}_B\mathbf{L}^{-\mathrm{T}}$, do the singular value decomposition

$$\mathbf{B} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}, \quad (18)$$

and compute $\mathbf{A}_{\mathrm{pre}} = \mathbf{U}^{\mathrm{T}}\mathbf{L}^{-1}$, $\mathbf{b}_{\mathrm{pre}} = -\mathbf{A}_{\mathrm{pre}}\boldsymbol{\mu}$, and

$$\mathbf{W}_{\mathrm{pre}} = [\,\mathbf{A}_{\mathrm{pre}}\,;\,\mathbf{b}_{\mathrm{pre}}\,] = \left[\,\mathbf{U}^{\mathrm{T}}\mathbf{L}^{-1}\,;\,-\mathbf{U}^{\mathrm{T}}\mathbf{L}^{-1}\boldsymbol{\mu}\,\right]. \quad (19)$$

The bias term $\mathbf{b}_{\mathrm{pre}}$ makes the data zero-mean on average, whereas the transformation $\mathbf{A}_{\mathrm{pre}}$ makes the average within-class covariance unity, and the between-class covariance diagonal. To revert the transformation by $\mathbf{W}_{\mathrm{pre}}$ we compute:

$$\mathbf{W}_{\mathrm{inv}} = \left[\,\mathbf{A}_{\mathrm{pre}}^{-1}\,;\,\boldsymbol{\mu}\,\right] = [\,\mathbf{L}\mathbf{U}\,;\,\boldsymbol{\mu}\,]. \quad (20)$$

If $\mathbf{W}$ is the transformation matrix in the original space, let $\mathbf{W}'$ be the equivalent transform in the space where the model and features are transformed by $\mathbf{W}_{\mathrm{pre}}$. In effect, $\mathbf{W}'$ is equivalent to transforming from the pre-transformed space to the original space with $\mathbf{W}_{\mathrm{inv}}$, applying $\mathbf{W}$, and then transforming back with $\mathbf{W}_{\mathrm{pre}}$:

$$\mathbf{W}' = \mathbf{W}_{\mathrm{pre}}\mathbf{W}^+\mathbf{W}_{\mathrm{inv}}^+ \quad (21)$$

$$\mathbf{W} = \mathbf{W}_{\mathrm{inv}}\mathbf{W}'^+\mathbf{W}_{\mathrm{pre}}^+ \quad (22)$$

where the notation $\mathbf{M}^+$ denotes a matrix $\mathbf{M}$ with an extra row whose last element is 1 and the rest are 0. If $\mathbf{P} \equiv \frac{\partial\mathcal{Q}(\mathbf{W})}{\partial\mathbf{W}}$ is the gradient with respect to the transform in the original space, we can compute the transformed gradient [6] as:

$$\mathbf{P}' = \mathbf{A}_{\mathrm{pre}}^{-\mathrm{T}}\mathbf{P}\mathbf{W}_{\mathrm{pre}}^{+\ \mathrm{T}}. \quad (23)$$

4.2. Hessian transform

Assuming that the model and data have been pre-transformed as described above, and approximating all variances with the average variance $\mathbf{I}$, the expected per-frame model likelihood is:

$$\mathcal{Q}(\mathbf{W}) = \log|\det\mathbf{A}| - \ldots \frac{1}{2}\sum_{j,i}\gamma_{ji}\mathcal{E}_{ji}\left[(\mathbf{A}\mathbf{x}+\mathbf{b}-\boldsymbol{\mu}_{ji})^{\mathrm{T}}(\mathbf{A}\mathbf{x}+\mathbf{b}-\boldsymbol{\mu}_{ji})\right], \quad (24)$$

where the expectation $\mathcal{E}_{ji}$ is over typical features $\mathbf{x}$ generated from Gaussian i of state j. Then we use the fact that the features $\mathbf{x}$ for Gaussian i of state j are distributed with unit variance and mean $\boldsymbol{\mu}_{ji}$, and the fact that the means $\boldsymbol{\mu}_{ji}$ have zero mean and variance $\boldsymbol{\Lambda}$ (obtained while computing the pre-transforms from Equation (18)), keeping only terms quadratic in $\mathbf{A}$ and/or $\mathbf{b}$, to get:

$$\mathcal{Q}(\mathbf{W}) = K + \log|\det\mathbf{A}| + \text{linear terms} - \frac{1}{2}\left[\mathrm{tr}\,(\mathbf{A}(\mathbf{I}+\boldsymbol{\Lambda})\mathbf{A}^{\mathrm{T}}) + \mathbf{b}^{\mathrm{T}}\mathbf{b}\right], \quad (25)$$

At this point it is possible to work out [6] that around $\mathbf{A} = \mathbf{I}$ and $\mathbf{b} = \mathbf{0}$,

$$\frac{\partial^2\mathcal{Q}}{\partial a_{rc}a_{r'c'}} = -\delta(r,c')\delta(c,r') - (1+\lambda_c)\delta(r,r')\delta(c,c') \quad (26)$$

$$\frac{\partial^2 \mathcal{Q}}{\partial a_{rc} b_{r'}} = 0 \tag{27}$$

$$\frac{\partial^2 \mathcal{Q}}{\partial b_r b_c} = -\delta(r,c), \tag{28}$$

where $\delta(\cdot,\cdot)$ is the Kronecker delta function. So the quadratic terms in a quadratic expansion of the auxiliary function around that point can be written as:

$$-\tfrac{1}{2}\sum_{r,c} a_{rc}a_{cr} + a_{rc}^2(1+\lambda_c) - \tfrac{1}{2}\sum_r b_r^2. \tag{29}$$

Note that the term $a_{rc}a_{cr}$ arises from the determinant and $a_{rc}^2(1+\lambda_c)$ arises from the expression $\mathrm{tr}\,(\mathbf{A}(\mathbf{I}+\mathbf{\Lambda})\mathbf{A}^{\mathrm{T}})$. This shows that each element of $\mathbf{A}$ is only correlated with its transpose, so with an appropriate reordering of the elements of $\mathbf{W}$, the Hessian would have a block-diagonal structure with blocks of size 2 and 1 (the size 1 blocks correspond to the diagonal of $\mathbf{A}$ and the elements of $\mathbf{b}$).

Consider the general problem where we have a parameter $\mathbf{f}$, and an auxiliary function of the form $\mathcal{Q}(\mathbf{f}) = \mathbf{f}\cdot\mathbf{g} - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{H}\mathbf{f}$, where $-\mathbf{H}$ is the Hessian w.r.t $\mathbf{f}$. We need to compute a co-ordinate transformation $\mathbf{f} \rightarrow \tilde{\mathbf{f}}$ such that the Hessian in the transformed space is $-\mathbf{I}$. By expressing $\mathbf{H}$ in terms of its Cholesky factors $\mathbf{H} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, it is clear that the appropriate transformed parameter is $\tilde{\mathbf{f}} = \mathbf{L}^{\mathrm{T}}\mathbf{f}$, since $\mathcal{Q}(\tilde{\mathbf{f}}) = \tilde{\mathbf{f}}\cdot(\mathbf{L}^{-1}\mathbf{g}) - \frac{1}{2}\tilde{\mathbf{f}}^{\mathrm{T}}\tilde{\mathbf{f}}$. This also makes clear that the appropriate transformation on the gradient is $\mathbf{L}^{-1}$.

The details of this Cholesky factor computation can be found in [6]. Multiplying $\mathbf{P}'$ (cf. equation (23)) with $\mathbf{L}^{-1}$ to obtain $\tilde{\mathbf{P}}$, has the following simple form. For $1 \le r \le d$ and $1 \le c < r$,

$$\tilde{p}_{rc} = (1+\lambda_c)^{-\frac{1}{2}} p'_{rc} \tag{30}$$

$$\tilde{p}_{cr} = -\left(1+\lambda_r - (1+\lambda_c)^{-1}\right)^{-\frac{1}{2}} (1+\lambda_c)^{-1} p'_{rc} + \left(1+\lambda_r - (1+\lambda_c)^{-1}\right)^{-\frac{1}{2}} p'_{cr} \tag{31}$$

$$\tilde{p}_{rr} = (2+\lambda_r)^{-\frac{1}{2}} p'_{rr} \tag{32}$$

$$\tilde{p}_{r,(d+1)} = p'_{r,(d+1)}, \tag{33}$$

where λ_i are the elements of the diagonal matrix $\mathbf{\Lambda}$ of (18).

In this transformed space, the proposed change $\mathbf{\Delta}$ in $\mathbf{W}$ will be:

$$\tilde{\mathbf{\Delta}} = \frac{1}{\beta}\tilde{\mathbf{P}}. \tag{34}$$

The factor $1/\beta$ is necessary because the expected Hessian is a per-observation quantity. The co-ordinate transformation from $\tilde{\mathbf{\Delta}}$ to $\mathbf{\Delta}'$ is as follows: for $1 \le r \le d$ and for $1 \le c < r$,

$$\delta'_{rc} = (1+\lambda_c)^{-\frac{1}{2}}\tilde{\delta}_{rc} - \left(1+\lambda_r - (1+\lambda_c)^{-1}\right)^{-\frac{1}{2}} (1+\lambda_c)^{-1}\tilde{\delta}_{cr} \tag{35}$$

$$\delta'_{cr} = \left(1+\lambda_r - (1+\lambda_c)^{-1}\right)^{-\frac{1}{2}} \tilde{\delta}_{cr} \tag{36}$$

$$\delta'_{rr} = (2+\lambda_r)^{-\frac{1}{2}}\tilde{\delta}_{rr} \tag{37}$$

$$\delta'_{r,(d+1)} = \tilde{\delta}_{r,(d+1)}. \tag{38}$$

We can then transform $\mathbf{\Delta}'$ to $\mathbf{\Delta}$; referring to Equation (22),

$$\mathbf{\Delta} = \mathbf{A}_{\mathrm{inv}}\mathbf{\Delta}'\mathbf{W}_{\mathrm{pre}}^{+}. \tag{39}$$

At this point a useful check is to make sure that in the three co-ordinate systems, the computed auxiliary function change based on a linear approximation is the same:

$$\mathrm{tr}\,(\mathbf{\Delta}\mathbf{P}^{\mathrm{T}}) = \mathrm{tr}\,(\mathbf{\Delta}'\mathbf{P}'^{\mathrm{T}}) = \mathrm{tr}\,(\tilde{\mathbf{\Delta}}\tilde{\mathbf{P}}^{\mathrm{T}}). \tag{40}$$

Algorithm 5.1 fMLLR estimation

Compute $\mathbf{W}_{\mathrm{pre}}$, $\mathbf{W}_{\mathrm{inv}}$ and $\mathbf{\Lambda}$ *// Eq. (18) – (20)*
Initialize: $\mathbf{W}_0^{(s)} = [\,\mathbf{I};\,\mathbf{0}\,]$
Accumulate statistics: *// Eq. (8) – (10)*
  $\beta \leftarrow \sum_{t,j}\gamma_j(t)$
  $\mathbf{K} \leftarrow \sum_{t,j}\gamma_j(t)\mathbf{\Sigma}_j^{-1}\boldsymbol{\mu}_j\mathbf{x}^{+}(t)^{\mathrm{T}}$
  $\mathbf{G}_j \leftarrow \sum_{t,j}\gamma_j(t)\mathbf{x}^{+}(t)\mathbf{x}^{+}(t)^{\mathrm{T}}$
for $n \leftarrow 1 \ldots N$ **do** *// e.g. for N=5 iterations*
  $\mathbf{G} \leftarrow \sum_j \mathbf{\Sigma}_j^{-1}\mathbf{W}_{n-1}^{(s)}\mathbf{G}_j$
  $\mathbf{P} \leftarrow \beta\left[\mathbf{A}_{n-1}^{-\mathrm{T}};\,\mathbf{0}\right] + \mathbf{K} - \mathbf{G}$
  $\mathbf{P}' \leftarrow \mathbf{A}_{\mathrm{inv}}^{\mathrm{T}}\mathbf{P}\mathbf{W}_{\mathrm{pre}}^{+\,\mathrm{T}}$ *// Pre-transform: Eq. (23)*
  $\mathbf{P}' \rightarrow \tilde{\mathbf{P}}$ *// Hessian transform: Eq. (30) – (33)*
  $\tilde{\mathbf{\Delta}} \leftarrow \frac{1}{\beta}\tilde{\mathbf{P}}$ *// Step in transformed space: Eq. (34)*
  $\tilde{\mathbf{\Delta}} \rightarrow \mathbf{\Delta}'$ *// Reverse Hessian transform: Eq. (35) – (38)*
  $\mathbf{\Delta} \leftarrow \mathbf{A}_{\mathrm{inv}}\mathbf{\Delta}'\mathbf{W}_{\mathrm{pre}}^{+}$ *// Reverse pre-transform: Eq. (39)*
  Compute step-size k *// Appendix A*
  $\mathbf{W}_n^{(s)} \leftarrow \mathbf{W}_{n-1}^{(s)} + k\mathbf{\Delta}$ *// Update*
end for

5. ADAPTATION RECIPE

In this section we summarize the steps for estimating the fMLLR transform in the form of a recipe (Algorithm 5.1).

The first step is calculating the pre-transform $\mathbf{W}_{\mathrm{pre}}$, the associated inverse transform $\mathbf{W}_{\mathrm{inv}}$, and the diagonal matrix of singular values $\mathbf{\Lambda}$. These quantities depend only on the models, and need to be computed only once before starting the iterative estimation of $\mathbf{W}$.

The next step is to initialize $\mathbf{W}^{(s)}$. If a previous estimate of $\mathbf{W}^{(s)}$ exists (for example, if we are running multiple passes over the adaptation data), it is used as the initial estimate. Otherwise $\mathbf{W}^{(s)} = [\,\mathbf{I}\,;\,\mathbf{0}\,]$ is a reasonable starting point.

For each pass over the adaptation data, we first accumulate the sufficient statistics β, $\mathbf{K}$, and $\mathbf{G}_i$, which can be done in $O(Td^2)$ time (type 1), or $O(TKd^2)$ time (type 2).

To iteratively update $\mathbf{W}^{(s)}$, we first compute the gradient $\mathbf{P}$ in the original space, and $\tilde{\mathbf{P}}$ in the fully transformed space. We then compute $\tilde{\mathbf{\Delta}}$, the change in $\mathbf{W}$ in this transformed space, from which we obtain the change $\mathbf{\Delta}$ in the original space by reversing the Hessian transform and the pre-transform.

We next compute the optimal step-size k using an iterative procedure described in Appendix A, which is then used to update $\mathbf{W}^{(s)}$:

$$\mathbf{W}^{(s)} \leftarrow \mathbf{W}^{(s)} + k\mathbf{\Delta}. \tag{41}$$

The time in update is dominated by Equation (13) (type 1) or (14) (type 2) which take time $O(Id^3)$ for the type 1 update and $O(Kd^3)$ in the type 2 update for a general type of basis (e.g. SPAM) but $O(Kd^2)$ for a "simple" basis as in standard diagonal or full-covariance system; this reduces to $O(d^3)$ for diagonal (versus $O(d^4)$ for the standard approach) and $O(d^4)$ for full-covariance (versus $O(d^4)$ in [3]). The accumulation time (the part proportional to T) is the same as standard approaches (type 2) or faster (type 1).

6. SUBSPACE VERSION OF FMLLR

We express Equation (2) in the fully transformed space:

$$\tilde{\mathbf{W}}^{(s)} = \tilde{\mathbf{W}}_0 + \sum_{b=1}^{B}\alpha_b^{(s)}\tilde{\mathbf{W}}_b. \tag{42}$$

	Number of SGMM Substates						
	1800	2700	4k	6k	9k	12k	16k
SGMM:	51.6	50.9	50.6	50.1	49.9	49.3	49.4
Per-speaker adaptation							
fMLLR:	49.7	49.4	48.7	48.3	48.0	47.6	47.6
Per-utterance adaptation							
fMLLR:	51.1	50.7	50.3	49.8	49.5	49.1	49.2
+ subspaces:	50.2	49.9	49.5	48.9	48.6	48.0	47.9

Table 1. fMLLR adaptation results (in % WER).

where $\tilde{\mathbf{W}}_b$ from an orthonormal basis, i.e., $\mathrm{tr}\,(\tilde{\mathbf{W}}_b \tilde{\mathbf{W}}_c^{\mathrm{T}}) = \delta(b, c)$.

With this subspace approach, Equation (34) is modified as:

$$\tilde{\boldsymbol{\Delta}} = \frac{1}{\beta} \sum_{b=1}^{B} \tilde{\mathbf{W}}_b \mathrm{tr}\,(\tilde{\mathbf{W}}_b \tilde{\mathbf{P}}^{\mathrm{T}}). \tag{43}$$

Note that in this method of calculation, the quantities $\alpha_b^{(s)}$ are *implicit* and are never referred to in the calculation, but the updated $\mathbf{W}$ will still be constrained by the subspace. This simplifies the coding procedure, but at the cost of slightly higher memory and storage requirement.

6.1. Training the bases

The auxiliary function improvement in the transformed space can be computed as $\frac{1}{2}\mathrm{tr}\,(\tilde{\boldsymbol{\Delta}}\tilde{\mathbf{P}}^{\mathrm{T}})$ (up to a linear approximation). This is the same as $\frac{1}{2}\mathrm{tr}\,(\frac{1}{\sqrt{\beta}}\tilde{\mathbf{P}}\frac{1}{\sqrt{\beta}}\tilde{\mathbf{P}}^{\mathrm{T}})$. So, the auxiliary function improvement is the trace of the scatter of $\frac{1}{\sqrt{\beta}}\tilde{\mathbf{P}}$ projected onto the subspace.

The first step in training the basis is to compute the quantity $\frac{1}{\sqrt{\beta}}\tilde{\mathbf{P}}^{(s)}$ for each speaker. We then compute the scatter matrix:

$$\mathbf{S} = \sum_s \mathrm{vec}\left(\frac{1}{\sqrt{\beta}}\tilde{\mathbf{P}}^{(s)}\right) \mathrm{vec}\left(\frac{1}{\sqrt{\beta}}\tilde{\mathbf{P}}^{(s)}\right)^{\mathrm{T}}, \tag{44}$$

where $\mathrm{vec}(\mathbf{M})$ represents concatenating the rows of a matrix $\mathbf{M}$ into a vector. The column vectors $\mathbf{u}_b$ corresponding to the top B singular values in the SVD of $\mathbf{S}$, $\mathbf{S} = \mathbf{U}\mathbf{L}\mathbf{V}^{\mathrm{T}}$, gives bases $\tilde{\mathbf{W}}_b$, i.e. $\mathbf{u}_b = \mathrm{vec}(\tilde{\mathbf{W}}_b)$.

7. EXPERIMENTS

Our experiments are with an SGMM style of system on the CALLHOME English database; see [4] for system details. Results are without speaker adaptive training.

In Table 1 we show adaptation results for different SGMM systems of varying model complexities [4]. We can see that the proposed method for fMLLR provides substantial improvements over an unadapted SGMM baseline when adapting using all the available data for a particular speaker. The improvements are consistent with those obtained by a standard implementation of fMLLR over a baseline system that uses conventional GMMs.

When adapting per-utterance (i.e. with little adaptation data), we see that normal fMLLR adaptation provides very modest gains (we use a minimum of 100 speech frames for adaptation, which gives good performance). However, using the subspace fMLLR with $B = 100$ basis transforms $\mathbf{W}_b$ (and the same minimum of 100 frames), we are able to get performance that is comparable to per-speaker adaptation.

8. CONCLUSIONS

In this paper we presented a novel estimation algorithm for fMLLR transforms with full-covariance models, which iteratively finds the gradient in a transformed space where the expected Hessian is proportional to unity. The proposed algorithm provides large improvements over a competitive unadapted SGMM baseline on an LVCSR task. It is also used to estimate a subspace-constrained fMLLR, which provides better results with limited adaptation data. The algorithm itself is independent of the SGMM framework, and can be applied to any HMM that uses GMM emission densities.

9. REFERENCES

[1] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, April 1998.

[2] K. Visweswariah, V. Goel, and R. Gopinath, "Structuring linear transforms for adaptation using training time information," in *Proc. IEEE ICASSP*, 2002, vol. 1, pp. 585–588.

[3] K. C. Sim and M. J. F. Gales, "Adaptation of precision matrix models on large vocabulary continuous speech recognition," in *Proc. IEEE ICASSP*, 2005, vol. I, pp. 97–100.

[4] D. Povey et al., "Subspace gaussian mixture models for speech recognition," *Submitted to ICASSP*, 2010.

[5] S. Axelrod et al., "Subspace constrained Gaussian mixture models for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 6, pp. 1144–1160, 2005.

[6] D. Povey, "A Tutorial-style introduction to Subspace Gaussian Mixture Models for Speech Recognition," Tech. Rep. MSR-TR-2009-111, Microsoft Research, 2009.

A. CALCULATING OPTIMAL STEP SIZE

The auxiliary function in the step size k is:

$$\begin{aligned}
\mathcal{Q}(k) &= \beta \log\det(\mathbf{A} + k\boldsymbol{\Delta}_{1:d,1:d}) \\
&\quad + k\,m - \tfrac{1}{2}k^2 n, && (45)\\
m &= \mathrm{tr}\,(\boldsymbol{\Delta}\mathbf{K}^{\mathrm{T}}) - \mathrm{tr}\,(\boldsymbol{\Delta}\mathbf{G}^{\mathrm{T}}) && (46)\\
n &= \sum_j \mathrm{tr}\left(\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Delta}\mathbf{G}_j\right) \quad \text{(type 1)} && (47)\\
n &= \sum_k \mathrm{tr}\left(\boldsymbol{\Delta}^{\mathrm{T}}\mathbf{A}_k\boldsymbol{\Delta}\mathbf{G}_k\right) \quad \text{(type 2)} && (48)
\end{aligned}$$

where $\boldsymbol{\Delta}_{1:d,1:d}$ is the first d columns of $\boldsymbol{\Delta}$. We use a Newton's method optimization for k. After computing

$$\begin{aligned}
\mathbf{B} &= (\mathbf{A} + k\boldsymbol{\Delta}_{1:d,1:d})^{-1}\boldsymbol{\Delta}_{1:d,1:d} && (49)\\
d_1 &= \beta\mathrm{tr}\,(\mathbf{B}) + m - kn && (50)\\
d_2 &= -\beta(\mathrm{tr}\,\mathbf{B}\mathbf{B}) - n && (51)
\end{aligned}$$

where d_1 and d_2 are the first and second derivatives of (45) with respect to k, we update k as:

$$\hat{k} = k - \frac{d_1}{d_2}. \tag{52}$$

At this point we check that $\mathcal{Q}(\hat{k}) \geq \mathcal{Q}(k)$. If $\mathcal{Q}(\cdot)$ decreases, we keep halving the step size $\hat{k} \leftarrow (k + \hat{k})/2$ until $\mathcal{Q}(\hat{k}) \geq \mathcal{Q}(k)$. The final k should typically be close to 1.