A factorized representation of FMLLR transform based on QR-decomposition

Shakti P. Rath, Martin Karafiát, Ondřej Glembek and Jan "Honza" Černocký

Brno University of Technology, Speech@FIT, Božetěchova 2, 612 66 Brno, Czech Republic.
rath@fit.vutbr.cz, karafiat@fit.vutbr.cz, glembek@fit.vutbr.cz, cernocky@fit.vutbr.cz

Abstract

In this paper, we propose a novel representation of the FMLLR transform. This is different from the standard FMLLR in that the linear transform (LT) is expressed in a factorized form such that each of the factors involves only one parameter. The representation is mainly motivated by QR-decomposition of a square matrix and hence is referred to as QR-FMLLR. The mathematical expressions and steps for maximum likelihood (ML) estimation of the parameters are presented. The ML estimation of QR-FMLLR does not require the use of numerical technique, such as gradient ascent, and it does not involve matrix inversion and computation of matrix determinant. On an LVCSR task, we show the performance of QR-FMLLR to be comparable to the standard FMLLR. We conjecture that QR-FMLLR is amenable to speaker adaptation using data that varies from very short to large and present a brief discussion on how this can be achieved.
Index Terms: FMLLR, QR Decomposition, Orthogonal Matrix, Givens Rotation, Upper Triangular Matrix

1. Introduction

For past several years, speaker adaptation has been one of the important fields of research in large vocabulary continuous speech recognition (LVCSR). It aims to reduce inter-speaker variability in speech by transforming the acoustic model parameters or the feature vectors. The dominant choices for speaker adaptation are maximum likelihood linear regression (MLLR) [1] and constrained MLLR (CMLLR) [2]. CMLLR is also known as feature space MLLR (FMLLR).

In FMLLR, a linear transform (LT) is applied on the feature vectors and the parameters of the LT are estimated from the speaker-specific data using maximum likelihood (ML). The gradient ascent algorithm is used for training. We will refer the approach proposed in [2] to as the standard FMLLR in the subsequent discussions.

In this paper, we propose a novel representation of the FMLLR transform. This is different from the standard FMLLR in that the LT is represented in a factorized form such that each of the factors involves only one parameter. More specifically, it is based on the observation that an $N \times N$ square matrix, in the present case, the FMLLR transform, can be expressed as a product of N^2 matrices (details are presented in Section 2), i.e.,

$$\mathbf{A} = \prod_{k=1}^{N^2} \mathbf{A}_k, \tag{1}$$

where $\mathbf{A}$ and $\mathbf{A}_k$'s are the FMLLR transform and the factors, respectively. It has the following properties:

- Each of the factors is $N \times N$ and *sparse*. Each factor involves *only one parameter* and is *full-rank*.
- Each factor operates either on a single feature component or on a 2-dimensional feature plane.

As it will be shown in Section 2, the presented factorized form is motivated by QR-decomposition of a square matrix and hence will be referred to as QR-FMLLR. QR-FMLLR can also be viewed as a structured LT. Other approaches to structured LT are [3, 4, 5].

One of the salient features of QR-FMLLR is that it does not require the use of numerical technique, such as gradient ascent, for estimation. Further, the estimation procedure does not involve matrix inversion and computation of matrix determinant. On an LVCSR task, we show the performance of QR-FMLLR to be comparable to the standard FMLLR.

In addition, we conjecture that QR-FMLLR is amenable to speaker adaptation with varying amount of speaker-specific data that ranges from very short to large. A brief discussion on this aspect of QR-FMLLR is presented in Section 5, which will be addressed in detail in our future work.

The rest of the paper is organized as follows. In Section 2, we present the novel representation of FMLLR, QR-FMLLR. The estimation procedure for QR-FMLLR is presented in Section 3. The experimental setups and results are discussed in Section 4. A future extension to QR-FMLLR is presented in Section 5. Finally, we conclude in Section 6.

2. Proposed factorized form of FMLLR

If N denotes the dimension of the feature vector, the $N \times N$ dimensional FMLLR transform can be expressed in the form shown in Eq. 1 in steps as follows:
Step 1: First, the LT is expressed as a product of an orthogonal matrix and an upper triangular matrix, i.e.,

$$\mathbf{A} = \mathbf{Q} \cdot \mathbf{R}\ ,\ s.t.\ \mathbf{Q}\mathbf{Q}^T = \mathbf{I} \text{ and } \mathbf{R} \text{ is Upper triangular} \tag{3}$$

where $\mathbf{I}$ denotes the identity matrix. This is motivated by QR-decomposition[1] [6] of a matrix.
Step 2: Then the orthogonal matrix, $\mathbf{Q}$, is expressed as a product of a series of Givens rotations [7, 6], which ensures orthogonality of the over-all matrix:

$$\mathbf{Q} = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{Q}(i, j, \theta_{ij}), \tag{4}$$

where $\mathbf{Q}(i, j, \theta_{ij})$ denotes the Givens rotation that rotates a feature vector on the 2-dimensional plane spanned by the co-ordinates (i, j) by an angle θ_{ij}. It leaves the feature components along all other co-ordinates un-altered. For instance, for

S. P. Rath was supported by "Detonation" project within SoMoPro - a program co-financed by South-Moravian region and EC under FP7 project No. 229603. The work was also partly supported by Technology Agency of the Czech Republic grant No. TA01011328, Czech Ministry of Education project No. MSM0021630528, and by European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

[1] This is not same as applying QR-decomposition to an existing LT.

$$\mathbf{Q}(2,4,\theta_{2,4}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos(\theta_{2,4}) & 0 & -\sin(\theta_{2,4}) & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \sin(\theta_{2,4}) & 0 & \cos(\theta_{2,4}) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{D}(2,d_2) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tilde{\mathbf{R}}(2,4,r_{2,4}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & r_{2,4} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$N = 5$ dimensional feature vector and $i = 2$ and $j = 4$, the Givens rotation has the structure shown in Eq 2. There is one such transform operating over every 2-dimensional plane.

$\mathbf{Q}(i,j,\theta_{ij})$ involves one parameter, θ_{ij}. It is sparse, full-rank, orthogonal with determinant +1. The product matrix, $\mathbf{Q}$, is also orthogonal with determinant +1. The number of factors (parameters) with $\mathbf{Q}$ is $\frac{N(N-1)}{2}$.

Step 3: The upper triangular matrix, $\mathbf{R}$, is first expressed as a product of a diagonal matrix, $\mathbf{D}$, and a purely upper triangular matrix (with 1's on the principal diagonal), $\tilde{\mathbf{R}}$. This ensures upper triangularity of the product matrix. Then, each of them is expressed as a product of single-parameter, sparse and full-rank matrices, i.e.,

$$\mathbf{R} = \mathbf{D} \cdot \tilde{\mathbf{R}} = \prod_{i=1}^{N} \mathbf{D}(i,d_i) \cdot \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \tilde{\mathbf{R}}(i,j,r_{ij}) \quad (5)$$

For instance, for $N = 5$, $i = 2$ and $j = 4$, $\mathbf{D}(i,d_i)$ and $\tilde{\mathbf{R}}(i,j,r_{ij})$ have the structures shown in Eq 2. $\mathbf{D}(i,d_i)$ scales only feature component i. On the other hand, $\tilde{\mathbf{R}}(i,j,r_{ij})$, similar to the Givens rotations, operates on the 2-dimensional plane (i,j). $\tilde{\mathbf{R}}(i,j,r_{ij})$'s and $\tilde{\mathbf{R}}$ are purely upper triangular with determinant +1. The number of factors in $\mathbf{R}$ is $N + \frac{N(N-1)}{2}$.

The parameters associated with the proposed representation, i.e., QR-FMLLR, are $\{\theta_{ij}\}$, $\{d_i\}$, $\{r_{ij}\}$ and the total number of parameters is N^2, which is same as the standard FMLLR.

2.1. Constructing LTs with different structures

In the frame-work of QR-FMLLR, LTs with various structures can be constructed by selectively removing some of the factors (parameters) from the full composition. For example, for 39 dimensional MFCC features, the static part of block diagonal LT can be constructed by allowing i to vary from 1 to 12 and j from $i+1$ to 13 in Eq. 4 and Eq. 5. Similarly, 5-diagonal LT can be constructed by varying i from 1 to 38 and j from $i+1$ to $i+2$. Orthogonal (or upper triangular) LT can be created by removing the upper triangular (or orthogonal) factor from the composition of $\mathbf{A}$. Experimental results with different LT structures are presented in Section 4.

3. Estimation of QR-FMLLR

In this section, we derive the expressions for ML estimation of the parameters of QR-FMLLR. The feature transformation model with FMLLR can be formulated as follows:

$$\mathbf{y}_t = \mathbf{A}\mathbf{x}_t + \mathbf{b} \quad (6)$$

where $\mathbf{A}$ is the linear part of the transform and $\mathbf{b}$ is the bias. $\mathbf{x}_t$ and $\mathbf{y}_t$ are the speaker un-normalized and the corresponding speaker normalized feature vectors, respectively. Let $\mathbf{X}$ and $\mathbf{Y}$ denote the sequences of original and speaker-normalized feature vectors, respectively. The likelihood function for the parameters, i.e., $\mathbf{A}$ and $\mathbf{b}$, w.r.t. $\mathbf{X}$ can be expressed as

$$\mathcal{L}(\mathbf{X}|\mathbf{A},\mathbf{b}) = \log\left|\frac{d\mathbf{Y}}{d\mathbf{X}}\right| + \log p\left(\mathbf{Y}|\,\lambda\right), \quad (7)$$

where λ is the HMM set and $|\cdot|$ denotes determinant of a matrix. The Jacobian, $\log\left|\frac{d\mathbf{Y}}{d\mathbf{X}}\right|$, appears due to feature transformation. Using the state and mixture sequences, $\mathbf{s}$ and $\mathbf{m}$, respectively, the EM auxiliary function for ML estimation is given by

$$\mathcal{Q} = \sum_{\mathbf{s},\mathbf{m}} p\left(\mathbf{s},\mathbf{m}|\mathbf{X},\lambda\right)\left(\log\left|\frac{d\mathbf{Y}}{d\mathbf{X}}\right| + \log p\left(\mathbf{Y},\mathbf{s},\mathbf{m}|\lambda\right)\right), \quad (8)$$

where the summation is taken over all state and mixture sequences. Noting that $\mathbf{y}_t$ depends only on $\mathbf{x}_t$ and ignoring the constant terms, the auxiliary function simplifies to

$$\mathcal{Q} = \sum_{j,m}\sum_{t=1}^{T} \gamma_{jm}(t)\left(\log\left|\frac{d\mathbf{y}_t}{d\mathbf{x}_t}\right| + \log\left(b_{jm}\left(\mathbf{y}_t\right)\right)\right), \quad (9)$$

where $\gamma_{jm}(t)$ is the posterior probability of mixture g_{jm} (mixture m in state j) in the speaker independent (SI) HMM at time t, $b_{jm}\left(\mathbf{y}_t\right)$ is the distribution of mixture g_{jm} and T is the number of frames from the speaker. We estimate $\mathbf{A}$ and $\mathbf{b}$ one after the other in an interleaved way. After arranging various terms, the auxiliary functions for $\mathbf{A}$ (for a known $\mathbf{b}$) and $\mathbf{b}$ (for a known $\mathbf{A}$) can be shown to be

$$\mathcal{Q}\left(\mathbf{A}|\mathbf{b}\right) = \beta\log|\mathbf{A}| + \mathrm{tr}\left\{\mathbf{A}\mathbf{K}_1{}^T\right\} - \frac{1}{2}\mathrm{vec}\left(\mathbf{A}\right)^T\mathbf{G}_1\mathrm{vec}\left(\mathbf{A}\right) - \mathrm{vec}\left(\mathbf{A}\right)^T\mathbf{G}_2\mathbf{b}, \quad (10)$$

$$\mathcal{Q}\left(\mathbf{b}|\mathbf{A}\right) = -\mathrm{vec}\left(\mathbf{A}\right)^T\mathbf{G}_2\mathbf{b} + \mathbf{k}_2^T\,\mathbf{b} - \frac{1}{2}\mathbf{b}^T\mathbf{G}_3\mathbf{b}, \quad (11)$$

where $\mathrm{vec}(\cdot)$ is the column-wise vectorization operation on the matrix and $\mathrm{tr}(\cdot)$ is the matrix trace operation. The statistics are defined in Appendix A. The ML estimation of $\mathbf{b}$ for a known $\mathbf{A}$ can be done in a straight-forward way by maximizing the corresponding auxiliary function. The ML estimation of $\mathbf{A}$, subject to the structure defined for QR-FMLLR, can be formulated as

$$\left\{\mathbf{Q}^*(\cdot),\mathbf{D}^*(\cdot),\tilde{\mathbf{R}}^*(\cdot)\right\} = \arg\max_{\left\{\mathbf{Q}(\cdot),\mathbf{D}(\cdot),\tilde{\mathbf{R}}(\cdot)\right\}} \mathcal{Q}\left(\mathbf{A}|\mathbf{b}\right), \quad (12)$$

where $\{\mathbf{Q}(\cdot),\mathbf{D}(\cdot),\tilde{\mathbf{R}}(\cdot)\}$ denotes the collection of all factors. Simultaneous estimation of all factors as shown in Eq. 12 would be mathematically intractable. Instead, we adopt a sequential estimation scheme, which is as follows:

We initialize the orthogonal ($\mathbf{Q}$), diagonal ($\mathbf{D}$) and purely upper-triangular ($\tilde{\mathbf{R}}$) matrices with the identity matrix and introduce the factors to the composition one at a time in a sequential manner. In each step, the parameter associated with the newly introduced factor is estimated. This is repeated for $\mathbf{Q}$, $\mathbf{D}$ and $\tilde{\mathbf{R}}$ one after the other. The matrices after k steps can be defined by the following recursive equations:

$$\mathbf{Q}_0 = \mathbf{D}_0 = \tilde{\mathbf{R}}_0 = \mathbf{I}, \quad (13)$$

$$\mathbf{Q}_k = \mathbf{Q}(i,j,\theta_{ij}) \cdot \mathbf{Q}_{k-1}\,, \quad (14)$$

$$\tilde{\mathbf{R}}_k = \tilde{\mathbf{R}}_{k-1} \cdot \tilde{\mathbf{R}}(i,j,r_{ij})\,, \quad (15)$$

$$\mathbf{D}_k = \mathbf{D}(i,d_i) \cdot \mathbf{D}_{k-1}. \quad (16)$$

Note that in this sequential estimation frame-work, each of the factors can be estimated using a separate invocation of EM, i.e., re-computing the statistics each time a new factor is introduced (Eq. 14-16) to the composition. However, it would require N^2-times processing of data and hence will be computationally expensive. Instead, we use the alignment of the original data, $\mathbf{X}$, and the corresponding statistics for estimation of all factors. The details are as follows:

Estimation of Givens rotation – $\mathbf{Q}(i,j,\theta_{ij})$: We pre-multiply the current estimate, $\mathbf{Q}_k$, of $\mathbf{Q}$ with the Givens rotation operating over the 2-dimensional plane, (i,j), to form the parametric representation of the matrix at step $k+1$, i.e.,

$$\mathbf{Q}_{k+1} = \mathbf{Q}(i,j,\theta_{ij}) \cdot \mathbf{Q}_k. \tag{17}$$

Similarly, the parametric form of $\mathbf{A}$ is constructed by

$$\mathbf{A}_{k+1} = \mathbf{Q}(i,j,\theta_{ij}) \cdot \mathbf{Q}_k \cdot \mathbf{D}_k \cdot \tilde{\mathbf{R}}_k. \tag{18}$$

Now, the only unknown parameter associated with $\mathbf{A}_{k+1}$ is θ_{ij}, which needs to be estimated. We noted that the trigonometric equation that will result while ML estimating θ_{ij} did not lead to a straight-forward solution. Using numerical technique it was observed that the estimates were small ($|\theta_{ij}| < 8$ degree) for all pairs of (i,j). This enabled us to use the small-angle approximation of the cos and sin functions, i.e.,

$$\cos(\theta_{i,j}) \approx 1 \;\text{ and }\; \sin(\theta_{i,j}) \approx \theta_{ij} \tag{19}$$

Such approximation converts the trigonometric equation to a polynomial one, which can be easily solved. As a consequence, however, the resulting $\mathbf{Q}(\cdot)$ and $\mathbf{Q}$ matrices will not be exactly orthogonal. Using the approximation and with some analysis, it can be shown that the EM auxiliary function for $\mathbf{A}_{k+1}$, which is equivalent to that for θ_{ij}, is given by

$$\mathcal{Q}\left(\mathbf{A}_{k+1}\right) \equiv \mathcal{Q}\left(\theta_{ij}\right) = \frac{1}{2} c_1 \theta_{ij}^2 + c_2 \theta_{ij}. \tag{20}$$

The constants c_1 and c_2 are defined in Appendix A. The maximum of Eq. 20 occurs at $\theta_{ij}^* = -c_2/c_1$. Then $\mathbf{Q}_k$ is updated using $\mathbf{Q}_{k+1}$=$\mathbf{Q}(i,j,\theta_{ij}^*)\mathbf{Q}_k$, which can be efficiently done as

$$\mathbf{q}_{p,k+1}^{\mathsf{r}} = \begin{cases} \mathbf{q}_{p,k}^{\mathsf{r}} - \theta_{ij}^* \, \mathbf{q}_{j,k}^{\mathsf{r}} & \text{if } p = i \\ \mathbf{q}_{p,k}^{\mathsf{r}} + \theta_{ij}^* \, \mathbf{q}_{i,k}^{\mathsf{r}} & \text{if } p = j \\ \mathbf{q}_{p,k}^{\mathsf{r}} & \text{otherwise} \end{cases}, \tag{21}$$

where $\mathbf{q}_{p,k+1}^{\mathsf{r}}$ denotes row p of $\mathbf{Q}_{k+1}$. Likewise, $\mathbf{A}_k$ is updated to $\mathbf{A}_{k+1}$, using similar equations:

$$\mathbf{a}_{p,k+1}^{\mathsf{r}} = \begin{cases} \mathbf{a}_{p,k}^{\mathsf{r}} - \theta_{ij}^* \, \mathbf{a}_{j,k}^{\mathsf{r}} & \text{if } p = i \\ \mathbf{a}_{p,k}^{\mathsf{r}} + \theta_{ij}^* \, \mathbf{a}_{i,k}^{\mathsf{r}} & \text{if } p = j \\ \mathbf{a}_{p,k}^{\mathsf{r}} & \text{otherwise} \end{cases}. \tag{22}$$

Estimation of purely upper triangular matrix – $\tilde{\mathbf{R}}(i,j,r_{ij})$: The purely upper triangular matrix, $\tilde{\mathbf{R}}_{k+1}$, is constructed by post-multiplying[2] the current estimate, $\tilde{\mathbf{R}}_k$, with $\tilde{\mathbf{R}}(i,j,r_{ij})$:

$$\tilde{\mathbf{R}}_{k+1} = \tilde{\mathbf{R}}_k \cdot \tilde{\mathbf{R}}(i,j,r_{ij}). \tag{23}$$

Similarly, $\mathbf{A}_{k+1}$ is constructed using

$$\mathbf{A}_{k+1} = \mathbf{Q}_k \cdot \mathbf{D}_k \cdot \tilde{\mathbf{R}}_k \cdot \tilde{\mathbf{R}}(i,j,r_{ij}). \tag{24}$$

Now, the unknown parameter to be estimated is r_{ij}. The EM auxiliary function for r_{ij} can be shown to be

$$\mathcal{Q}\left(\mathbf{A}_{k+1}\right) \equiv \mathcal{Q}\left(r_{ij}\right) = \frac{1}{2} c_3 r_{ij}^2 + c_4 r_{ij}, \tag{25}$$

where c_3 and c_4 are defined in Appendix A. The ML estimate of r_{ij} is $r_{ij}^* = -c_4/c_3$. The column p of $\tilde{\mathbf{R}}_{k+1}$ is given by

$$\tilde{\mathbf{r}}_{p,k+1}^{\mathsf{c}} = \begin{cases} \tilde{\mathbf{r}}_{p,k}^{\mathsf{c}} + r_{ij}^* \, \tilde{\mathbf{r}}_{i,k}^{\mathsf{c}} & \text{if } p = j \\ \tilde{\mathbf{r}}_{p,k}^{\mathsf{c}} & \text{otherwise} \end{cases}. \tag{26}$$

$\mathbf{A}_k$ is updated to $\mathbf{A}_{k+1}$ using similar equation.

Estimation of diagonal matrix – $\mathbf{D}(i,d_i)$: For diagonal case, $\mathbf{D}_{k+1}$ and $\mathbf{A}_{k+1}$ are constructed as follows

$$\mathbf{D}_{k+1} = \mathbf{D}(i,d_i) \cdot \mathbf{D}_k \;,\; \mathbf{A}_{k+1} = \mathbf{Q}_k \cdot \mathbf{D}(i,d_i) \cdot \mathbf{D}_k \cdot \tilde{\mathbf{R}}_k \tag{27}$$

[2]Post-multiplication makes the analysis simpler.

Table 1: WER (%) on CTS task with standard LT structures including bias. Diag=Diagonal

Structure of Matrix	Num. of parameters	QR-FMLLR	standard FMLLR
No Adaptation	-	43.0	43.0
Full	1521	39.6	39.8
Block Diag (BD)	507	39.9	40.0
Diag	39	41.8	42.3

Now d_i is the unknown and the EM auxiliary function is

$$\mathcal{Q}(d_i) = \beta \log|d_i| + \frac{1}{2} c_5 \left(d_i - 1\right)^2 + c_6 \left(d_i - 1\right). \tag{28}$$

The constants c_5 and c_6 are defined in Appendix A. The ML estimate of d_i, d_i^*, can be obtained by solving

$$\frac{\partial\, \mathcal{Q}(d_i)}{\partial\, d_i} = \beta \frac{1}{d_i} + c_5 \left(d_i - 1\right) + c_6 = 0. \tag{29}$$

The updated $\mathbf{A}_{k+1}$ is given by

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \left(d_i^* - 1\right) \mathbf{q}_{i,k}^{\mathsf{c}} \, \tilde{\mathbf{r}}_{i,k}^{\mathsf{r}}. \tag{30}$$

Since the estimate of one parameter depends on the values of the other parameters[3], it is necessary to repeat the estimation of all parameters a few times. Iterating 5 times gave satisfactory results in our experiments. After each iteration, bias is updated.

4. Experimental setup and Results

For experiments, the speaker independent (SI) baseline model was trained on the *ctstrain-04* training set, which is a subset of the *h5train-03* set defined at the University of Cambridge. The training set contains about 278 hours of speech from Switchboard I, II and Call Home English. Test was done on the *Hub5 Eval-01* test set, which was used during NIST 2001 CTS evaluation. It consists of 3 subsets of 20 conversations from Switchboard-1, Switchboard-2 and Switchboard cellular corpora and contains more than 6 hours of speech. Bi-gram language models from AMI speech recognition system for NIST Rich Transcriptions 2007 was used during decoding [8]. 39-dimensional MFCC features that consist of 13 (C_1 to C_{12} and C_0) static, Δ and Δ^2 components were used. Speaker-wise cepstral mean and cepstral variance normalization were performed both during training and test. 3-state cross-word triphone HMM models with 20 mixtures per state were used. There were approximately 148000 Gaussian mixtures and 7369 independent states in the HMM set. The test set included data from 120 speakers. The duration of test data per speaker was 3 minutes in average. The un-supervised mode of speaker adaptation was used, where the first-pass transcription was used for alignments. Speaker adaptive training [2] was not used.

Percentage of word error rates (WER) with QR-FMLLR and FMLLR for full, block diagonal (BD) and diagonal matrices are presented in Table 1. Please refer to Section 2.1 for a description on how to create the structured QR-FMLLRs. It can be observed from the Table that the performance of QR-FMLLR is comparable to FMLLR for full and block diagonal matrices and is better than FMLLR for the diagonal case.

Using QR-FMLLR, LTs with various other structures can be constructed (Section 2.1). In Table 2, the WER with structures such as upper triangular, orthogonal, diagonal with different widths, are presented. BD + $S\Delta$+ΔS (or, BD + $S\Delta^2$+$\Delta^2 S$) in the Table indicates that the LT has block structure, where along with the blocks at the static-static, Δ-Δ and

[3]Please refer to the constants defined in Appendix A. The dependency also holds with standard FMLLR.

Table 2: WER (%) with other LT structures: UT = Upper Triangular, OR = Orthogonal, BD = Block diagonal, $S\Delta$ = Block at static-Δ position, ΔS = Block at Δ-static position, $S\Delta^2$ = Block at static-Δ^2 position, $\Delta^2 S$ = Block at Δ^2-static position

Structure of Matrix	Num. of parameters	QR-FMLLR
UT	780	41.3
OR	741	40.4
OR + Diagonal	780	39.9
BD + $S\Delta$+ΔS	845	39.8
BD + $S\Delta^2$+$\Delta^2 S$	845	39.8
5-Diagonal	189	40.7
3-Diagonal	115	40.8

Δ^2-Δ^2 positions, two additional blocks are used at the static-Δ and Δ-static (or, static-Δ^2 and Δ^2-static) positions. The following observations can be made:

• The performance of orthogonal transform along with a diagonal transform (OR + Diagonal) gives performance comparable to block diagonal (BD) transform (shown in Table 1).

• Although an upper triangular (UT) LT involves more parameters than the BD transform, its performance is worse than BD.

• The performance of 3-diagonal and 5-diagonal matrices are comparable and both are worse than BD transform.

• BD + $S\Delta$+ΔS and BD + $S\Delta^2$+$\Delta^2 S$ give marginal improvements over BD.

5. Controlling structure of LT using MDL

In the frame-work of QR-FMLLR, LTs with number of parameters varying between 0 to N^2 can be constructed, where zero parameter corresponds to the identity matrix (i.e., no adaptation) and N^2 parameters to the full LT. This can be achieved by removing some of the factors from the composition of the full LT. This effectively reduces the number of speaker-specific parameters to be estimated from the adaptation data and hence would require less data from the speaker for reliable estimation.

For a given data set, the optimum number of parameters (factors), k^*, and the optimal composition of the LT with k^* factors can be obtained as follows. The optimum k can be obtained using minimum description length (MDL) [9]. The MDL objective function deduced for QR-FMLLR is given by:

$$\mathcal{Q}^{\texttt{MDL}}(k) = -\log \mathcal{Q}(\mathbf{A}_k^*) + \frac{k}{2}\log\beta + \log N^2 \qquad (31)$$

where $\mathbf{A}_k^*$ denotes the ML estimate of $\mathbf{A}$ with k parameters, $\mathcal{Q}(\mathbf{A}_k^*)$ is the corresponding value of EM auxiliary function and β is the number of frames in the adaptation set. The MDL objective function needs to be minimized to obtain the optimum k. On the other hand, the optimal composition of the LT with k factors can be decided by searching over all (allowed) k-tuples of factors. Both the steps can be integrated into a single frame-work by appropriately modifying the sequential estimation of QR-FMLLR and including an MDL-based stopping criteria.

Hence, QR-FMLLR along with MDL provides a data-driven frame-work to optimally adjust the structure of the LT. We believe, this will be useful for speaker adaptation when data available from the speakers vary from very short to large. We will investigate on this topic in our future work.

6. Conclusions

In this paper, a factorized representation of FMLLR, i.e., QR-FMLLR, is proposed. The mathematical expressions and steps for ML estimation are presented. It was shown that using QR-FMLLR, LTs with various structures, such as full, orthogonal, upper triangular, 5-diagonal and block-diagonal can be constructed. The speaker adaptation results are presented on an LVCSR task. Our future extension to QR-FMLLR is discussed.

A. Statistics and constants

The statistics used in Eq. 10 and 11 are as follows:

$$\beta = \sum_{jm}\sum_{t=1}^{T}\gamma_{jm}(t), \mathbf{K}_1 = \sum_{jm}\sum_{t=1}^{T}\gamma_{jm}(t)\boldsymbol{\Sigma}_{jm}^{-1}\boldsymbol{\mu}_{jm}\mathbf{x}_t^T,$$

$$\mathbf{G}_1 = \sum_{jm}\sum_{t=1}^{T}\gamma_{jm}(t)\ \mathbf{x}_t\mathbf{x}_t^T \otimes \boldsymbol{\Sigma}_{jm}^{-1}, \mathbf{k}_2 = \sum_{jm}\sum_{t=1}^{T}\gamma_{jm}(t)\boldsymbol{\Sigma}_{jm}^{-1}\boldsymbol{\mu}_{jm},$$

$$\mathbf{G}_2 = \sum_{jm}\sum_{t=1}^{T}\gamma_{jm}(t)\ \mathbf{x}_t \otimes \boldsymbol{\Sigma}_{jm}^{-1}, \mathbf{G}_3 = \sum_{jm}\sum_{t=1}^{T}\gamma_{jm}(t)\ \boldsymbol{\Sigma}_{jm}^{-1}.$$

where $\otimes$ denotes the Kronecker product. $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{\Sigma}_{jm}$ are the mean and co-variance matrix of mixture g_{jm} (mixture m in state j) in the HMM set, respectively. $\gamma_{jm}(t)$ is the posterior probability of mixture g_{jm} at time t and T is the number of frames from the speaker. Let

$$\mathbf{k}_{ij} = \begin{bmatrix} \mathbf{K}_1^{(i)} \\ \mathbf{K}_1^{(j)} \end{bmatrix},\ \mathbf{g}_{ij}^k = \begin{bmatrix} \mathbf{G}_1^{(i,:)}\ \text{vec}\,(\mathbf{A}_k) + \mathbf{G}_2\ \mathbf{b} \\ \mathbf{G}_1^{(j,:)}\ \text{vec}\,(\mathbf{A}_k) + \mathbf{G}_2\ \mathbf{b} \end{bmatrix},$$

$$\mathbf{G}_{ij} = \begin{bmatrix} \mathbf{G}_1^{(i,i)} & ; & \mathbf{G}_1^{(i,j)} \\ \mathbf{G}_1^{(j,i)} & ; & \mathbf{G}_1^{(j,j)} \end{bmatrix},$$

where the super-scripted $\mathbf{G}_1^{(i,j)}$ denotes selecting the (i,j)-th sub-matrix of $\mathbf{G}_1$ and $\mathbf{K}_1^{(i)}$ denotes i-th column of $\mathbf{K}_1$. The constants appearing in Eq. 20, 25 and 28 are as follows:

$$c_1 = \left[-\mathbf{a}_{i,k}^{\text{r}}; -\mathbf{a}_{j,k}^{\text{r}}\right]\left(\mathbf{k}_{ij} - \mathbf{g}_{ij}^k\right) - \left[-\mathbf{a}_{j,k}^{\text{r}}; \mathbf{a}_{i,k}^{\text{r}}\right]\mathbf{G}_{ij}\left[-\mathbf{a}_{j,k}^{\text{r}}; \mathbf{a}_{i,k}^{\text{r}}\right]^T$$

$$c_2 = \left[-\mathbf{a}_j^{\text{r}}; \mathbf{a}_i^{\text{r}}\right]\left(\mathbf{k}_{ij} - \mathbf{g}_{ij}^k\right), c_3 = -(\mathbf{a}_{i,k}^{\text{c}})^T\mathbf{G}_1^{(j,j)}\mathbf{a}_{i,k}^{\text{c}},$$

$$c_4 = (\mathbf{a}_{i,k}^{\text{c}})^T\left\{\mathbf{K}_1^{(j)} - \mathbf{G}_1{}^{(j,:)}\text{vec}\,(\mathbf{A}_k) - \mathbf{G}_2\mathbf{b}\right\},$$

$$c_5 = -\text{vec}\left(\mathbf{q}_{i,k}^{\text{c}}\,\tilde{\mathbf{r}}_{i,k}^{\text{r}}\right)^T\mathbf{G}_1\ \text{vec}\left(\mathbf{q}_{i,k}^{\text{c}}\,\tilde{\mathbf{r}}_{i,k}^{\text{r}}\right),$$

$$c_6 = \text{vec}\left(\mathbf{q}_{i,k}^{\text{c}}\,\tilde{\mathbf{r}}_{i,k}^{\text{r}}\right)^T\left\{\text{vec}\,(\mathbf{K}_1) - \mathbf{G}_1\text{vec}\,(\mathbf{A}_k) - \mathbf{G}_2\mathbf{b}\right\}.$$

B. References

[1] C. J. Leggetter, "Improved Acoustic Modelling for HMMs Using Linear Transformations," Ph.D. dissertation, University of Cambridge, UK, 1995.

[2] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

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

[4] G. H. Ding, B. Xu, J. I. Sipila, and Y. Cao, "Fast speaker adaptation using triple diagonal and shared block diagonal transform matrices," in *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, 2003, pp. 300–303.

[5] D. Povey and K. Yao, "A basis method for robust estimation of constrained MLLR," in *Proc. of ICASSP*, Prague, 2011.

[6] G. H. Golub and C. F. V. Loan, "Matrix computations," in *Johns Hopkins University Press (3rd Edition)*, 1996.

[7] W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular format," *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, no. 1, pp. 26–50, 1958.

[8] T. Hain and L. Burget et al., "The AMIDA 2009 meeting transcription system," in *Proc. Interspeech*, vol. 2010, no. 9, Makuhari, Chiba, JP, 2010, pp. 358–361.

[9] K. Shinoda and T. Watanabe, "Speaker adaptation with autonomous model complexity control by MDL principle," in *Proc. of ICASSP*, Atlanta, May 1996.