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, Božetčeova 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) 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.,

\[
A = \prod_{k=1}^{N^2} A_k
\]

where \( A \) and \( A_k \)’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 long. 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.,

\[
A = Q \cdot R, \text{ s.t. } QQ^T = I \text{ and } R \text{ is Upper triangular}
\]

where \( I \) denotes the identity matrix. This is motivated by QR-decomposition of a matrix.

**Step 2:** Then the orthogonal matrix, \( Q \), is expressed as a product of a series of Givens rotations [7][6], which ensures orthogonality of the over-all matrix:

\[
Q = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} Q(i, j, \theta_{ij}),
\]

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

\footnote{This is not same as applying QR-decomposition to an existing LT.}
$N = 5$ 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.

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

**Step 3**: The upper triangular matrix, $R$, is first expressed as a product of a diagonal matrix, $D$, and a purely upper triangular matrix (with $1$’s on the principal diagonal), $\tilde{R}$. 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.,

\[
R = D \cdot \tilde{R} = \prod_{i=1}^{N} D(i, d_i) \cdot \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \tilde{R}(i, j, r_{ij})
\]

(5)

For instance, for $N = 5$, $i = 2$ and $j = 4$, $D(i, d_i)$ and $\tilde{R}(i, j, r_{ij})$ have the structures shown in Eq. 2, $D(i, d_i)$ scales only feature component $i$. On the other hand, $\tilde{R}(i, j, r_{ij})$, similar to the Givens rotations, operates on the 2-dimensional plane $(i, j)$. $\tilde{R}(i, j, r_{ij})$’s and $R$ are purely upper triangular with determinant $+1$. The number of factors in $R$ 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. 2 and Eq. 3. 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 $A$. Experimental results with different LT structures are presented in Section 3.

### 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:

\[
y_t = AX_t + b
\]

(6)

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

\[
L(X; A, b) = \log \left| \frac{dY}{dX} \right| + \log p(Y|\lambda),
\]

(7)

where $\lambda$ is the HMM set and $|\cdot|$ denotes determinant of a matrix. The Jacobian, $\log |\frac{dY}{dX}|$, appears due to feature transformation.

Using the state and mixture sequences, $s$ and $m$, respectively, the EM auxiliary function for ML estimation is given by

\[
Q = \sum_{s,m} p(s, m|X, \lambda) \log \left( \frac{dY}{dX} \right) + \log p(Y, s, m|\lambda),
\]

(8)

where the summation is taken over all state and mixture sequences. Noting that $Y_s$ depends only on $X_t$ and ignoring the constant terms, the auxiliary function simplifies to

\[
Q = \sum_{j} \sum_{m} \gamma_j m(t) \left( \log \frac{dY_t}{dx_t} + \log (b_{jm}(y_t)) \right),
\]

(9)

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

\[
Q(A|b) = \beta \log |A| + tr \left( A^T G_1 A \right) - \frac{1}{2} vec(A)^T G_2 vec(A) + \log p(A|\lambda),
\]

(10)

\[
Q(b|A) = - \log vec(A)^T G_2 b + \frac{1}{2} b^T G_3 b - \frac{1}{2} tr(G_4),
\]

(11)

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

\[
\{Q^*(\cdot), D^*(\cdot), R^*(\cdot)\} = \arg \max Q(A|b),
\]

(12)

where $\{Q(\cdot), D(\cdot), R(\cdot)\}$ 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 ($Q$), diagonal ($D$) and purely upper-triangular ($R$) 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 $Q$, $D$, and $R$, one after the other. The matrices after $k$ steps can be defined by the following recursive equations:

\[
Q_0 = D_0 = R_0 = I,
\]

(13)

\[
Q_k = Q(i, j, \theta_{ij}) \cdot Q_{k-1},
\]

(14)

\[
R_k = R_{k-1} \cdot R(i, j, r_{ij}),
\]

(15)

\[
D_k = D(i, d_i) \cdot D_{k-1}.
\]

(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, $X_t$ 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.,
\[
\mathbf{Q}_{k+1} = \mathbf{Q}(i, j, \theta_{ij}) \cdot \mathbf{Q}_k.
\]
(17)
Similarly, the parametric form of \( \mathbf{A} \) is estimated by
\[
\mathbf{A}_{k+1} = \mathbf{Q}(i, j, \theta_{ij}) \cdot \mathbf{Q}_k \cdot \mathbf{D}_k \cdot \mathbf{R}_k.
\]
(18)
Now, the only unknown parameter associated with \( \mathbf{A}_{k+1} \) is \( \theta_{ij} \), which needs to be estimated. We noted that the trigonometric equation that will result while ML estimating \( \theta_{ij} \) did not lead to a straightforward solution. Using numerical technique it was observed that the estimates were small (\(|\theta_{ij}| < 8\) degrees) for all pairs of \((i,j)\). This enabled us to use the small-angle approximation of the cos and sin functions, i.e.,
\[
\cos(\theta_{ij}) \approx 1 \quad \text{and} \quad \sin(\theta_{ij}) \approx \theta_{ij}
\]
(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 \( \theta_{ij} \), is given by
\[
\mathbf{Q}(\mathbf{A}_{k+1}) \equiv \mathbf{Q}(\theta_{ij}) = \frac{1}{2} \left( c_i \theta_{ij}^2 + c_d \theta_{ij} \right).
\]
(20)
The constants \( c_i \) and \( c_d \) are defined in Appendix [A]. The maximum of Eq. (20) occurs at \( \theta_{ij}^* = -c_d/c_i \). 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
\[
\mathbf{Q}^*_{p,k+1} = \begin{cases} \mathbf{Q}^*_{p,k} - \theta_{ij}^* \mathbf{Q}^*_{q,k} & \text{if } p = i \\ \mathbf{Q}^*_{p,k} + \theta_{ij}^* \mathbf{Q}^*_{q,k} & \text{if } p = j \\ \mathbf{Q}^*_{p,k} & \text{otherwise} \end{cases}
\]
(21)
where \( \mathbf{Q}^*_{p,k+1} \) denotes row \( p \) of \( \mathbf{Q}_{k+1} \). Likewise, \( \mathbf{A}_k \) is updated to \( \mathbf{A}_{k+1} \), using similar equations:
\[
\mathbf{A}^*_{p,k+1} = \begin{cases} \mathbf{A}^*_{p,k} - \theta_{ij}^* \mathbf{A}^*_{q,k} & \text{if } p = i \\ \mathbf{A}^*_{p,k} + \theta_{ij}^* \mathbf{A}^*_{q,k} & \text{if } p = j \\ \mathbf{A}^*_{p,k} & \text{otherwise} \end{cases}
\]
(22)
**Estimation of purely upper triangular matrix – \( \hat{\mathbf{R}}(i, j, r_{ij}) \):** The purely upper triangular matrix, \( \hat{\mathbf{R}}_{k+1} \), is constructed by post-multiplying the current estimate, \( \mathbf{R}_k \), with \( \hat{\mathbf{R}}(i, j, r_{ij}): \)
\[
\hat{\mathbf{R}}_{k+1} = \mathbf{R}_k \cdot \hat{\mathbf{R}}(i, j, r_{ij}).
\]
(23)
Similarly, \( \mathbf{A}_{k+1} \) is constructed using
\[
\mathbf{A}_{k+1} = \mathbf{Q}_k \cdot \mathbf{D}_k \cdot \mathbf{R}_k \cdot \hat{\mathbf{R}}(i, j, r_{ij}).
\]
(24)
Now, the unknown parameter to be estimated is \( r_{ij} \). The EM auxiliary function for \( r_{ij} \) can be shown to be
\[
\mathbf{Q}(\mathbf{A}_{k+1}) \equiv \mathbf{Q}(r_{ij}) = \frac{1}{2} \left( d_i r_{ij}^2 + d_d r_{ij} \right),
\]
(25)
where \( d_i \) and \( d_d \) are defined in Appendix [A]. The ML estimate of \( r_{ij} \) is \( r_{ij}^* = -c_d/c_i \). The column \( p \) of \( \hat{\mathbf{R}}_{k+1} \) is updated by
\[
\hat{\mathbf{R}^*}_{p,k+1} = \begin{cases} \hat{\mathbf{R}^*}_{p,k} + r_{ij}^* \hat{\mathbf{R}^*}_{q,k} & \text{if } p = j \\ \hat{\mathbf{R}^*}_{p,k} & \text{otherwise} \end{cases}
\]
(26)
\( \mathbf{A}_k \) is updated to \( \mathbf{A}_{k+1} \) using similar equation.

**Estimation of diagonal matrix – \( \hat{\mathbf{D}}(d, d_i) \):** For diagonal case, \( \mathbf{D}_{k+1} \) and \( \mathbf{A}_{k+1} \) are constructed as follows
\[
\mathbf{D}_{k+1} = \mathbf{D}(d, d_i) \cdot \mathbf{D}_k \quad \mathbf{A}_{k+1} = \mathbf{Q}_k \cdot \mathbf{D}(d, d_i) \cdot \mathbf{D}_k \cdot \mathbf{R}_k
\]
(27)

<table>
<thead>
<tr>
<th>Structure of Matrix</th>
<th>Num. of parameters</th>
<th>QR-FMLLR</th>
<th>standard FMLLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Adaptation</td>
<td>43.0</td>
<td>43.0</td>
<td></td>
</tr>
<tr>
<td>Full</td>
<td>1521</td>
<td>39.6</td>
<td>39.8</td>
</tr>
<tr>
<td>Block Diag (BD)</td>
<td>507</td>
<td>39.9</td>
<td>40.0</td>
</tr>
<tr>
<td>Diag</td>
<td>39</td>
<td>41.8</td>
<td>42.3</td>
</tr>
</tbody>
</table>

Now \( d_i \) is the unknown and the EM auxiliary function is
\[
\mathbf{Q}(d_i) = \beta \log |d_i| + \frac{1}{c_i} \left( d_i - 1 \right)^2 + c_d \left( d_i - 1 \right).
\]
(28)
The constants \( c_2 \) and \( c_d \) are defined in Appendix [A]. The ML estimate of \( d_i, d_{ij}^* \), can be obtained by solving
\[
\frac{\partial \mathbf{Q}(d_i)}{\partial d_i} = \beta - c_d + c_2 (d_i - 1) + c_d = 0.
\]
(29)
The updated \( \mathbf{A}_{k+1} \) is given by
\[
\mathbf{A}_{k+1} = \mathbf{A}_k + (d_{ij} - 1) \mathbf{Q}^*_{ij,k} \mathbf{R}^*_{ij,k}.
\]
(30)
Since the estimate of one parameter depends on the values of the other parameter\(^3\), 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 cstrain-04 training set, which is a subset of the hstrain-03 set defined at the University of Cambridge. The training set contains about 278 hours of speech from Switchboard 1, 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, \( \Delta \) and \( \Delta^2 \) 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] 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 + \( S \Delta + \Delta S \) (or, BD + \( S \Delta^2 + \Delta S^2 \)) in the Table indicates that the LT has block structure, where along with the blocks at the static-static, \( \Delta \Delta \) 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, SΔ = Block at static-Δ position, ΔS = Block at Δ-static position, SΔ2 = Block at static-Δ2 position, Δ2S = Block at Δ2-static position

<table>
<thead>
<tr>
<th>Structure of Matrix</th>
<th>Num. of Parameters</th>
<th>QR-FMLLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>UT</td>
<td>780</td>
<td>41.3</td>
</tr>
<tr>
<td>OR</td>
<td>741</td>
<td>40.4</td>
</tr>
<tr>
<td>OR + Diagonal</td>
<td>780</td>
<td>39.9</td>
</tr>
<tr>
<td>BD + SΔ + ΔS</td>
<td>845</td>
<td>39.8</td>
</tr>
<tr>
<td>BD + SΔ2 + ΔS</td>
<td>845</td>
<td>39.8</td>
</tr>
<tr>
<td>5-Diagonal</td>
<td>189</td>
<td>40.7</td>
</tr>
<tr>
<td>3-Diagonal</td>
<td>115</td>
<td>40.8</td>
</tr>
</tbody>
</table>

$β$ is the number of frames in the adaptation set. The MDL obtained using minimum description length (MDL) [9]. The MDL difference between two matrices is defined as

$$∆ = \frac{1}{2} \log β + \log \text{det}(A) - \log \text{det}(B)$$

where $A$ and $B$ are the two matrices. The MDL difference is obtained by subtracting the log of the determinant of the two matrices. The MDL difference is a measure of the difference in the information content of the two matrices.

### 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) [2]. The MDL objective function deduced for QR-FMLLR is given by:

$$Q_{ML}(k) = -\log Q(\lambda_k^*) + \frac{k}{2} \log β + \log N^2$$

where $\lambda_k^*$ denotes the ML estimate of $\lambda$ with $k$ parameters, $Q(\lambda_k^*)$ 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 framework 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 framework 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:

$$\beta = \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma_{jm}(t), \quad K_1 = \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma_{jm}(t) \mu_{jm} x_t^T,$$

$$G_1 = \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma_{jm}(t) x_t x_t^T \mu_{jm}^{-1}, \quad K_2 = \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma_{jm}(t) \mu_{jm}^{-1},$$

$$G_2 = \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma_{jm}(t) x_t \mu_{jm}^{-1}, \quad G_3 = \sum_{t=1}^{T} \sum_{j=1}^{N} \gamma_{jm}(t) \mu_{jm}^{-1},$$

where $\otimes$ denotes the Kronecker product. $\mu_{jm}$ and $\Sigma_{jm}$ are the mean and co-variance matrix of mixture $g_{jm}$ (mixture in state $j$) in the HMM set, respectively. $\gamma_{jm}(t)$ is the posterior probability of mixture $g_{jm}$ at time $t$ and $N$ is the number of frames from the speaker. Let $k_{ij} = \begin{bmatrix} K_1^{(i)} & K_1^{(j)} \end{bmatrix}, \quad g_{ij} = \begin{bmatrix} G_1^{(i)} & G_1^{(j)} \end{bmatrix}, \quad G_{ij} = \begin{bmatrix} G_1^{(i)} & G_1^{(j)} \end{bmatrix}$, where the super-scripted $G_{ij}^{(i-j)}$ denotes selecting the $(i,j)$-th sub-matrix of $G_1$ and $K_1^{(i)}$ denotes $i$-th column of $K_1$. The constants appearing in Eq. [20] and [28] are as follows:

$$c_1 = -a_{ij}^T \left( a_{ij} - \mu_{ij} \right) + \frac{1}{2} a_{ij}^T G_1^{(i-j)} a_{ij},$$

$$c_2 = -a_{ik}^T \left( a_{ik} - \mu_{ik} \right), \quad c_3 = -a_{ik}^T G_1^{(i-j)} a_{ik},$$

$$c_4 = \left( \mu_{ik} \right)^T \left( K_1^{(i)} - G_1^{(i-j)} \right) \mu_{ik} - G_1^{(i-j)} \mu_{ik},$$

$$c_5 = \left( \mu_{ik} \right)^T \left( K_1^{(i)} - G_1^{(i-j)} \right) \mu_{ik} - G_1^{(i-j)} \mu_{ik},$$

$$c_6 = \left( \mu_{ik} \right)^T \left( K_1^{(i)} - G_1^{(i-j)} \right) \mu_{ik} - G_1^{(i-j)} \mu_{ik}.$$

### B. References


