TWO-STAGE SPEAKER ADAPTATION IN SUBSPACE GAUSSIAN MIXTURE MODELS

Sina Hamidi Ghalehjegh and Richard C. Rose

Department of Electrical and Computer Engineering, McGill University, Montreal, Canada

sina.hamidighalehjegh@mail.mcgill.ca, rose@ece.mcgill.ca

ABSTRACT

A two-stage speaker adaptation approach is proposed for the subspace Gaussian mixture model (SGMM) [1] in large vocabulary automatic speech recognition (ASR). The SGMM differs from the more well known continuous density hidden Markov model (CDHMM) in that a large portion of the SGMM parameters are dedicated to shared full covariance Gaussian subspace parameters and a relatively small number of parameters are used for state dependent projection vectors. Both model space and feature space adaptation are investigated. First, an efficient regression based approach for subspace vector adaptation (SVA) is presented. Second, an efficient approach is presented for feature space adaptation using constrained maximum likelihood linear regression (CMLLR) in the SGMM. While both of these adaptation scenarios have previously been investigated in the context of the SGMM [2, 3], a more efficient and numerically stable procedure is presented here for estimating the parameters of the regression based transformations. Both transformation matrices are obtained using an optimization technique that iteratively updates the rows of the regression matrices. It is shown that using these feature space and model space approaches for unsupervised speaker adaptation provides complementary improvements in SGMM based ASR word accuracy.

Index Terms— Speaker adaptation, Phonetic subspace, Constrained MLLR, Row-by-row update

1. INTRODUCTION

This paper presents an efficient optimization approach for estimating regression based adaptation parameters in the subspace Gaussian mixture model (SGMM) [1]. This approach is applied to unsupervised speaker adaptation in large vocabulary ASR. The SGMM is an alternative acoustic modeling technique to the continuous density hidden Markov model (CDHMM). SGMM parameters represent a globally shared model subspace which is trained from data to capture principal directions of phonetic variability using low dimensional state-dependent vectors referred to as "state projection vectors". A large portion of the SGMM parameters are dedicated to shared full covariance Gaussian subspace parameters and a relatively small number of parameters are used for state projection vectors. In practice, the SGMM facilitates acoustic modeling with a smaller amount of training data. This is thought to result partly from the subspace constraints provided by the model structure and partly due to the overall reduction in the total number of model parameters [4].

The estimation of both feature space and model space linear regression matrices for speaker adaptation is addressed in the context of the SGMM acoustic model. First, a model space approach is described in Section 3 where a linear regression based transformation is used for unsupervised adaptation of state projection vectors to new speakers. This approach, referred to as subspace vector adaptation (SVA), was motivated in [2] by studies which demonstrated an empirical relationship between SGMM state projection vectors and articulatory information in speech [2, 5]. It was shown in [2] that, while SVA resulted in a substantial improvement in word accuracy (WAC) in multiple task domains, the full matrix update solution for the regression matrix is both numerically unstable and computationally expensive. To deal with this issue, a more stable and efficient row-by-row update method is presented in Section 4.1 where the SVA regression matrix is estimated by iteratively updating its rows.

The second speaker adaptation approach investigated here involves feature space adaptation using constrained maximum likelihood linear regression (CMLLR) in the SGMM. Applying a linear transformation in the feature space is widely known to be effective for speaker adaptation in the CDHMM framework [6–8]. CMLLR was applied in [3] to feature space adaptation in the SGMM. Solution for the maximum likelihood estimate of the optimum CMLLR transformation matrix is complicated in the case of the SGMM by the fact that the underlying subspaces are represented by full covariance Gaussian densities. A repeated line search in the direction of the gradient was used in [3] resulting in a computationally expensive solution as is discussed in Section 3. A simpler and more efficient procedure for obtaining the CMLLR transformation matrix for the SGMM using a row-by-row update method is presented in Section 4.2.

The key to simple and efficient maximum likelihood estimation of both SVA and CMLLR transformations is the row-by-row update algorithms presented in Section 4. These ap-

This work was supported by the Nuance Foundation and the Quebec Government MDEIE.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

proaches are similar to that used by Sim et. al. in [9] for adapting precision matrix models in the CDHMM. The results of an experimental study presented in Section 5 demonstrate that complementary improvements in ASR word accuracy are obtained by combining these two approaches in an unsupervised speaker adaptation scenario.

2. SUBSPACE GAUSSIAN MIXTURE MODEL

This section provides a summary of the SGMM acoustic model [1]. The observation densities in an SGMM system with J states for a feature vector $\mathbf{x}(t)$ in state j is given as:

$$b_j(\mathbf{x}(t)) = \sum_{i=1}^{I} \omega_{ji} \mathcal{N}(\mathbf{x}(t)|\boldsymbol{\mu}_{ji}, \boldsymbol{\Sigma}_i). \quad (1)$$

The means, $\boldsymbol{\mu}_{ji}$, and mixture weights, ω_{ji}, are controlled by a global mapping from a vector space, through a single "state projection vector," to the GMM parameters space. The full-covariance matrices, $\boldsymbol{\Sigma}_i$, are shared among all the HMM states.

The i^{th} mean vector for state j is a projection into the i^{th} subspace defined by a $S \times S$ linear subspace matrix $\mathbf{M}_i$ as $\boldsymbol{\mu}_{ji} = \mathbf{M}_i \mathbf{v}_j$. The $S \times 1$ state projection vectors, $\mathbf{v}_j$, are the state specific parameters in the SGMM. The weights, ω_{ji}, in (1) are obtained from the $\mathbf{v}_j$ using a log-linear model:

$$\omega_{ji} = \frac{\exp \boldsymbol{\omega}_i^T \mathbf{v}_j}{\sum_{i'=1}^{I} \exp \boldsymbol{\omega}_{i'}^T \mathbf{v}_j} \quad (2)$$

where $\boldsymbol{\omega}_i$ denotes the weight projection vector for the i^{th} subspace. A more detatiled discussion of the SGMM model parameterization and parameter estimation is provided in [1].

3. SGMM SPEAKER ADAPTATION

This section presents the objective functions that are optimized for full matrix estimation of SVA and CMLLR regression transformations in the SGMM [2, 3]. The numerical issues associated with these full matrix estimates are also discussed and motivation is provided for the row-by-row solutions presented in Section 4.

3.1. Subspace Vector Adaptation (SVA)

This section provides a brief description of the subspace vector adaptation (SVA) introduced in [2]. An affine transformation is defined for state project vector adaptation:

$$\hat{\mathbf{v}}_j = \mathbf{A}\mathbf{v}_j + \mathbf{b} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{v}_j \\ 1 \end{bmatrix} = \mathbf{W}\mathbf{u}_j \quad (3)$$

in which $\mathbf{A}$ is a $S \times S$ matrix and $\mathbf{b}$ is a $S \times 1$ vector. The transformation matrix $\mathbf{W}$ is found in a maximum likelihood (ML) fashion. The required auxiliary function is as:

$$\mathcal{Q}(\mathbf{W}) = \sum_j \mathbf{f}_j^T \mathbf{W}\mathbf{u}_j - 0.5 \sum_j \mathbf{u}_j^T \mathbf{W}^T \mathbf{C}_j \mathbf{W}\mathbf{u}_j \quad (4)$$

where

$$\mathbf{f}_j = \sum_i (\gamma_{ji} - \gamma_j \bar{\bar{\omega}}_{ji} + \max(\gamma_j \bar{\bar{\omega}}_{ji}, \gamma_{ji}) \boldsymbol{\omega}_i^T \bar{\bar{\mathbf{W}}} \mathbf{u}_j) \boldsymbol{\omega}_i + \mathbf{y}_j$$

$$\mathbf{C}_j = \sum_i \left(\max(\gamma_j \bar{\bar{\omega}}_{ji}, \gamma_{ji}) \boldsymbol{\omega}_i \boldsymbol{\omega}_i^T + \gamma_{ji} \mathbf{H}_i \right) \quad (5)$$

are the necessary statistics that need to be accumulated [2]. The summation over j in (4) is computed for the states associated with a given regression class. The $\bar{\bar{\omega}}_{ji}$ and $\bar{\bar{\mathbf{W}}}$ correspond to their current values and

$$\begin{aligned} \mathbf{H}_i &= \mathbf{M}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{M}_i \\ \mathbf{y}_j &= \textstyle\sum_{t,i} \gamma_{ji}(t) \mathbf{M}_i^T \boldsymbol{\Sigma}_i^{-1} \mathbf{x}^T(t) \\ \gamma_{ji} &= \textstyle\sum_t \gamma_{ji}(t). \end{aligned} \quad (6)$$

In (6), $\gamma_{ji}(t)$ is the probability of being in state j and Gaussian mixture component i at time t and $\mathbf{X} = \{\mathbf{x}(1), \ldots, \mathbf{x}(T)\}$ is the sequence of the adaptation feature vectors on which the transformation is to be trained. Getting a closed form solution for the transformation parameters requires inverting a low-rank $S^2 \times S^2$ matrix which is computationally expensive and causes numerical instability. Therefore, [2] takes a gradient ascent approach to overcome these issues. To do so, a closed form solution is obtained for the bias term $\mathbf{b}$ and a gradient ascent technique is obtained for the matrix $\mathbf{A}$:

$$\mathbf{A}^{(k)} = \mathbf{A}^{(k-1)} + \lambda^{(k-1)} \left. \frac{\partial \mathcal{Q}(\mathbf{W})}{\partial \mathbf{A}} \right|_{\mathbf{A}^{(k-1)}} \quad (7)$$

where $\frac{\partial \mathcal{Q}(\mathbf{W})}{\partial \mathbf{A}} = \sum_j \mathbf{f}_j \mathbf{v}_j^T - \sum_j \mathbf{C}_j \mathbf{W}\mathbf{u}_j \mathbf{v}_j^T$. Nevertheless, this update method itself raises some other problems. First, $\mathbf{b}$ and $\mathbf{A}$ are updated separately, which might introduce some estimation mismatch. Second, the gradient matrix in (7) needs to be computed at each update iteration. This is clearly computationally expensive. Third, using a single step size is not a reasonable choice as different rows of $\mathbf{A}$ might need to be updated with different step sizes. As a result, it is difficult to obtain a reliable and efficient estimate of the transformation matrix using this approach. To solve these issues, we propose a new row-by-row update method in Section 4.1 in which $\mathbf{W}$ is estimated by iteratively updating its rows. This method not only provides a very simple update procedure, but is also numerically well-behaved.

3.2. Constrained MLLR (CMLLR)

This section provides a brief description of the constrained maximum likelihood linear regression (CMLLR) technique introduced in [3]. This method uses an affine transformation of the form:

$$\hat{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ 1 \end{bmatrix} = \mathbf{W}\boldsymbol{\zeta}(t) \quad (8)$$

in which $\mathbf{x}(t)$ is the adaptation data at time t. The required auxiliary function is as:

$$\mathcal{Q}(\mathbf{W}) = \beta \log |\det \mathbf{A}| + \text{tr}(\mathbf{W}\bar{\mathbf{K}}^T) - 0.5 \textstyle\sum_i \text{tr}(\mathbf{W}\mathbf{G}_i\mathbf{W}^T\boldsymbol{\Sigma}_i^{-1}) \quad (9)$$

where

$$\begin{aligned} \beta &= \sum_{t,j,i} \gamma_{ji}(t) \\ \bar{\mathbf{K}} &= \sum_{t,j,i} \gamma_{ji}(t) \boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_{ji} \boldsymbol{\zeta}(t)^T \\ \mathbf{G}_i &= \sum_{t,j} \gamma_{ji}(t) \boldsymbol{\zeta}(t) \boldsymbol{\zeta}(t)^T \end{aligned} \quad (10)$$

and summation over t is computed for the set of frames that is associated with a given regression class [3]. To maximize (9) w.r.t. $\mathbf{W}$, a repeated line search is performed in the direction of gradient [3]. To do so, a pre-transformation is used so that the expected Hessian matrix is proportional to identity matrix. Nevertheless, this algorithm is computationally expensive and does not have an efficient closed form solution. To overcome these problems, we propose a simple and efficient row-by-row update method in Section 4.2 in which $\mathbf{W}$ is estimated by iteratively updating its rows.

4. ROW-BY-ROW APPROACH

4.1. Row-By-Row Approach to SVA

A new optimization method is proposed here for finding the SVA transformation matrix. Starting from (4), one can re-write the auxiliary function in terms of the rows of $\mathbf{W}$:

$$\mathcal{Q}(\mathbf{W}) = \sum_p \mathbf{w}_p \mathbf{k}_p^T - 0.5 \sum_{l,p} \mathbf{w}_l \mathbf{F}^{(l,p)} \mathbf{w}_p^T \quad (11)$$

in which $\mathbf{w}_p$ and $\mathbf{k}_p$ are the p^{th} rows of $\mathbf{W}$ and $\mathbf{K}$, respectively, and $\mathbf{K}$ and $\mathbf{F}^{(l,p)}$ are defined as follow:

$$\begin{aligned} \mathbf{F}^{(l,p)} &= \sum_j \mathbf{u}_j \mathbf{u}_j^T c_j^{(l,p)} \\ \mathbf{K} &= \sum_j \mathbf{f}_j \mathbf{u}_j^T \end{aligned} \quad (12)$$

where $c_j^{(l,p)}$ is the $(l,p)^{\text{th}}$ element of matrix $\mathbf{C}_j$. In order to optimize (11) w.r.t. the r^{th} row of $\mathbf{W}$, we assume that the remaining rows are fixed to their current values. By re-ordering (11) w.r.t. $\mathbf{w}_r$ and discarding the other terms, one can define the *row-wise* auxiliary function as:

$$\mathcal{Q}(\mathbf{w}_r) = \mathbf{w}_r (\mathbf{k}_r^T - \sum_{p \neq r} \mathbf{F}^{(r,p)} \mathbf{w}_p^T) - 0.5 \mathbf{w}_r \mathbf{F}^{(r,r)} \mathbf{w}_r^T. \quad (13)$$

Differentiating (13) w.r.t. $\mathbf{w}_r$ and equating it with zero:

$$\hat{\mathbf{w}}_r = (\mathbf{k}_r - \sum_{p \neq r} \mathbf{w}_p \mathbf{F}^{(r,p)}) \mathbf{F}^{(r,r)^{-1}}. \quad (14)$$

It is apparent that this update formula is dependent on the other rows through the summation in (14). Therefore, an initial estimate of $\mathbf{W} = [\mathbf{A}\ \mathbf{b}]$ is required and an iterative approach is utilized. Although $\mathbf{A}$ and $\mathbf{b}$ can be initialized as an identity matrix and a zero vector, respectively, a better starting point can be achieved by using a diagonal $\mathbf{A}$ approximation and using (14). Multiple epochs are run over the adaptation data. Four epochs have been found to be sufficient. Algorithm 1 summarizes the row-by-row approach to SVA.

Algorithm 1: Row-by-row update approach to SVA

for *epoch=1* **to** *4* **do**
- Accumulate Statistics:
  - Do Forward-Backward to get $\gamma_{ji}(t)$
  - Compute $\mathbf{H}_i$, $\mathbf{y}_j$ and γ_{ji} as (6)
  - Compute $\mathbf{C}_j$ and $\mathbf{f}_j$ as (5)
  - Compute $\mathbf{K}$ and $\mathbf{F}^{(l,p)}$ as (12)
- Initialize:
  - $\mathbf{W}_0 = [\mathbf{I}, \mathbf{0}]$
  - Using diagonal assumption, use (14) to obtain a better initialization for $\mathbf{W}$
- **for** *iteration=1* **to** *10* **do**
  - **for** *row=1* **to** *S* **do**
    - Use (14) to find a new estimate: $\hat{\mathbf{w}}_r$
    - **if** $\mathcal{Q}(\hat{\mathbf{w}}_r) > \mathcal{Q}(\mathbf{w}_r)$ **then**
      - update the row
- Update $\mathbf{v}_j$ as (3)

4.2. Row-By-Row Approach to CMLLR

A row-by-row technique is proposed for estimating the CMLLR transformation matrix which is similar to that of Section 4.1. Re-writing (9) in terms of rows of $\mathbf{W}$:

$$\mathcal{Q}(\mathbf{W}) = \beta \log |\det \mathbf{A}| + \sum_p \mathbf{w}_p \bar{\mathbf{k}}_p^T - 0.5 \sum_{l,p} \mathbf{w}_l \bar{\mathbf{F}}^{(l,p)} \mathbf{w}_p^T \quad (15)$$

where $\mathbf{w}_p$ and $\bar{\mathbf{k}}_p$ are the p^{th} rows of $\mathbf{W}$ and $\bar{\mathbf{K}}$, respectively. In (15), $\bar{\mathbf{F}}^{(l,p)}$ is defined as :

$$\bar{\mathbf{F}}^{(l,p)} = \sum_i \mathbf{G}_i \sigma_i^{(l,p)} \quad (16)$$

where $\sigma_i^{(l,p)}$ is the $(l,p)^{\text{th}}$ element of $\boldsymbol{\Sigma}_i^{-1}$. In order to optimize (15) w.r.t. the r^{th} row of W, we assume that the remaining rows are fixed to their current values. By re-ordering (15) and just keeping terms dependent on $\mathbf{w}_r$, one can define the *row-wise* auxiliary function as:

$$\mathcal{Q}(\mathbf{w}_r) = \beta \log |\mathbf{p}_r \mathbf{w}_r^T| + \mathbf{w}_r (\bar{\mathbf{k}}_r^T - \sum_{p \neq r} \bar{\mathbf{F}}^{(r,p)} \mathbf{w}_p^T) - 0.5 \mathbf{w}_r \bar{\mathbf{F}}^{(r,r)} \mathbf{w}_r^T$$

where $\mathbf{p}_r = [\ p_{r1}\ \ \dots\ \ p_{rn}\ \ 0\ ]$ is the extended cofactor vector in which $p_{ij} = \text{cofactor}(\mathbf{A}_{ij})$. Taking the derivative w.r.t. $\mathbf{w}_r$ and equating with zero, we can find the optimum row vector:

$$\hat{\mathbf{w}}_r = (\alpha \mathbf{p}_r + \bar{\mathbf{k}}_r - \sum_{p \neq r} \mathbf{w}_p \bar{\mathbf{F}}^{(r,p)}) \bar{\mathbf{F}}^{(r,r)^{-1}} \quad (17)$$

and α is the root of equation defined as:

$$\alpha^2 \mathbf{p}_r \bar{\mathbf{F}}^{(r,r)^{-1}} \mathbf{p}_r^T + \alpha \mathbf{p}_r \bar{\mathbf{F}}^{(r,r)^{-1}} (\bar{\mathbf{k}}_r^T - \sum_{p \neq r} \bar{\mathbf{F}}^{(r,p)} \mathbf{w}_p^T) - \beta = 0.$$

This is similar to the procedure followed in [8]. This update formula is dependent on the other rows through the summation in (17). Therefore, an initial estimate of $\mathbf{W} = [\mathbf{A}\ \mathbf{b}]$ is

required and an iterative approach is utilized. We initialize $\mathbf{A}$ and $\mathbf{b}$ as an identity matrix and a zero vector, respectively[1]. Multiple epochs are run over the adaptation data. Six epochs have been found to be sufficient. Algorithm 2 summarizes the row-by-row optimization approach to CMLLR.

Algorithm 2: Row-by-row update approach to CMLLR

for *epoch=1* **to** *6* **do**
- Accumulate Statistics:
  - Do Forward-Backward to get $\gamma_{ji}(t)$
  - Compute β, $\bar{\mathbf{K}}$ and $\mathbf{G}_i$ as (10)
  - Compute $\bar{\mathbf{F}}^{(l,p)}$ as (16)
- Initialize: $\mathbf{W}_0 = [\mathbf{I}, \mathbf{0}]$
- **for** *iteration=1* **to** *10* **do**
  - **for** *row=1* **to** *S* **do**
    - Use (17) to find a new estimate: $\hat{\mathbf{w}}_r$
    - **if** $\mathcal{Q}(\hat{\mathbf{w}}_r) > \mathcal{Q}(\mathbf{w}_r)$ **then**
      - update the row
- Transform feature vectors as (8)

5. EXPERIMENTAL STUDY

This section presents an experimental study evaluating the performance of SVA and CMLLR adaptation techniques. Performance is reported as the word error rate (WER) on the resource management (RM) speech corpus. CDHMM training is done using the standard HTK toolkit [10] and SGMM training is done using an extended version of HTK [4].

The RM training corpus consists of 3990 utterances taken from the RM SI-109 training set. The 39 dimensional feature vectors consists of 12 MFCCs, normalized energy and the first and second differences. The baseline CDHMM system uses three-state left-to-right HMM triphone models. Decision tree clustering was used to obtain a system with 1704 states, each having 6 Gaussian mixtures per state. The choice of the number of Gaussians per state was obtained by observing the performance of the CDHMM system on the test set. The SGMM system was trained using the same training data set with $I = 256$ Gaussian mixtures shared between 1704 states. ASR WER was evaluated using 1200 utterances from 12 speakers taken from the RM speaker dependent evaluation (SDE) set. Recognition was performed using the standard RM 991 word bi-gram language model.

Table 1 displays the WERs for all the systems evaluated in the study. The baseline SGMM is shown to provide a reduction in WER of approximately 10% relative to the baseline CDHMM. Speaker adaptation experiments are performed in an *unsupervised* mode with an average duration of 5.33 minutes of data per speaker by using two regression classes of `speech` and `silence`. The second row of Table 1 shows that CMLLR adaptation applied to the CDHMM provides a 32% reduction in WER relative to the CDHMM baseline WER. Rows four and five of Table 1 show the ASR result for SVA by using "full matrix" and "row-by-row" methods, respectively. The row-by-row update method significantly improves the ASR performance while reducing the computational complexity by an order of 4. The sixth row shows that CMLLR adaptation obtained using the row-by-row update method provides 30% reduction in WER relative to the SGMM baseline. We speculate that a similar reduction in complexity might be seen by comparing the full matrix method given in [3] and our proposed row-by-row update method in Section 4.2.

Table 1. The ASR WER for different unsupervised speaker adaptation scenarios

System	WER [%]
Baseline CDHMM	4.91
CDHMM+CMLLR	3.33
Baseline SGMM	4.52
SGMM+SVA (full matrix)	3.90
SGMM+SVA	3.51
SGMM+CMLLR	3.15
SGMM+SVA+CMLLR	3.09
SGMM+CMLLR+SVA	**2.89**

In the last experiment, we use SVA and CMLLR approaches in a complementary mode. Row eight of Table 1 displays the WER when CMLLR feature space adaptation is performed followed by SVA model space adaptation for the SGMM. It is clear from the table that combining feature space and model space speaker adaptation provides complementary improvement in the recognition performance.

6. CONCLUSION

A new optimization method was proposed for estimating the adaptation parameters in subspace vector adaptation and constrained MLLR speaker adaptation techniques in the SGMM framework. In both cases, the transformation matrices were obtained by iteratively updating their rows. The experimental studies were done for RM speech corpus. We observed that the row-by-row update method not only provides an efficient estimate of the transformation matrices, but also greatly reduces the computational complexity. We finally observed that combining SVA and CMLLR adaptation techniques in the SGMM framework in a complementary mode can provide 36% relative reduction in the WER with respect to the SGMM baseline. Future work will investigate the application of proposed method to other less numerically stable EM based parameter estimation and adaptation, for example, dealing with acoustic mismatch for training multilingual SGMM [11, 12].

[1]Unlike the SVA case, diagonal $\mathbf{A}$ approximation for initialization does not give any further recognition improvement.

7. REFERENCES

[1] D. Povey et al., "The subspace Gaussian mixture model – a structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.

[2] S. Hamidi Ghalehjegh and R. Rose, "Phonetic subspace adaptation for automatic speech recognition," in *ICASSP*, 2013, pp. 7937 – 7941.

[3] A. Ghoshal et al., "A novel estimation of feature-space MLLR for full-covariance models," in *ICASSP*, 2010, pp. 4310–4313.

[4] R. Rose, S. C. Yin, and Y. Tang, "An investigation of subspace modeling for phonetic and speaker variability in automatic speech recognition," in *ICASSP*, 2011, pp. 4508–4511.

[5] L. Burget et al., "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in *ICASSP*, 2010, pp. 4334–4337.

[6] A. Sankar and C. H. Lee, "Robust speech recognition based on stochastic matching," in *ICASSP*, 1995, pp. 121–124.

[7] V. V. Digalakis, D. Ritschev, and L.G. Neumeyer, "Speaker adaptation using constrained estimation of Gaussian mixtures," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 357–366, 1995.

[8] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech and language*, vol. 12, pp. 75–98, 1997.

[9] K. C. Sim and M. J. F. Gales, "Adaptation of precision matrix models on large vocabulary continuous speech recognition," in *Proc. ICASSP*, 2005, vol. 1, pp. 97–100.

[10] S. Young et al., "The HTK book (for HTK version 3.4)," 2006.

[11] A. Mohan, S. Hamidi Ghalehjegh, and R. C. Rose, "Dealing with acoustic mismatch for training multilingual subspace Gaussian mixture models for speech recognition," in *ICASSP*. IEEE, 2012, pp. 4893–4896.

[12] A. Mohan, R. C. Rose, S. Hamidi Ghalehjegh, and S. Umesh, "Acoustic modeling for speech recognition in Indian languages in an agricultural commodities task domain," *Speech Communication*, vol. 56, pp. 167–180, 2014.