INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Subspace LHUC for Fast Adaptation of Deep Neural Network Acoustic Models

Lahiru Samarakoon, Khe Chai Sim

National University of Singapore
lahiruts@comp.nus.edu.sg, simkc@comp.nus.edu.sg

Abstract

Recently, the learning hidden unit contributions (LHUC) method is proposed for the adaptation of deep neural network (DNN) based acoustic models for automatic speech recognition (ASR). In LHUC, a set of speaker dependent (SD) parameters is estimated to linearly recombine the hidden units in an unsupervised fashion. Although LHUC performs considerably well, the gains diminish when the availability of the adaptation data amount decreases. Moreover, the per-speaker footprint of LHUC adaptation is in thousands and it is not desirable. Therefore, in this work, we propose the subspace LHUC, where the SD parameters are estimated in a subspace and connected to various layers through a new set of adaptively trained weights. We evaluate the subspace LHUC in the Aurora4 and AMI IHM tasks. Experimental results show that the subspace LHUC outperforms standard LHUC adaptation. With utterance-level fast adaptation, the subspace LHUC achieved 11.3% and 4.5% relative improvements over the standard LHUC for the Aurora4 and AMI IHM tasks respectively. Furthermore, the subspace LHUC reduces the per-speaker footprint by 94% over the standard LHUC adaptation.

Index Terms: Automatic Speech Recognition, Speaker Adaptation, LHUC.

1. Introduction

All machine learning techniques including DNNs are susceptible to performance degradations due to the mismatches between the training and testing conditions. The mismatch causing variabilities can be normalized by transforming the model to match testing conditions or by augmenting the runtime features to match the model. In ASR, speaker adaptation techniques are used to minimize the mismatch between the training and testing conditions due to the speaker variability.

Maximum a posteriori (MAP) [1] and maximum likelihood linear regression (MLLR) [2, 3] are commonly used to adapt GMM-hidden markov model (HMM) systems. In addition, speaker adaptive training (SAT) has been applied to GMM-HMM systems [4, 5]. Speaker adaptation for DNNs is important as it reduces error rates significantly [6, 7, 8, 9, 10, 11, 12]. However, it is difficult to interpret DNNs as meaningful structures as it is possible for GMMs. Therefore, DNN adaptation is challenging, especially when performed with a small amount of data in an unsupervised fashion. A popular approach for combining the GMM-HMM adaptation techniques with the DNNs is to train the tandem systems [13, 14, 15]. In tandem systems, a DNN is trained to extract bottleneck features to train a GMM-HMM system. Furthermore, In [16], temporally varying weight regression (TVWR) framework [17] is used to combine DNN and GMM acoustic models to improve the ASR robustness.

DNN Adaptation techniques can be categorized into two broad approaches: test-only adaptation (simply refers to as adaptation), and adaptive training. Adaptation methods reduce the mismatch by changing a well-trained model to match the test condition, whereas adaptive training reduces the mismatch during training. The adaptation methods can be categorized into 3 classes: linear transformation based adaptation, subspace or subset adaptation and regularized adaptation. Linear transformation based adaptation methods augment the original DNN model with a condition dependent linear layer [18, 19, 20, 21, 22, 23]. In subspace or subset methods, the adaptation is performed to a subset of model parameters or on a pruned model [24, 25, 26, 27, 28, 29, 30, 31]. Regularization based adaptation helps to perform the adaptation more conservatively [8, 32]. Adaptive training methods can be categorized into cluster adaptive training (CAT) [33, 34], feature normalization techniques like CMLLR [3], vocal tract length normalization (VTLN) [35], and speaker-aware training (SaT) [9, 10, 6].

In this paper, we propose the subspace LHUC method which performs the LHUC [31] adaptation in a subspace. In LHUC, the adaptation data is used to estimate a set of condition dependent parameters for the adapting condition to linearly recombine the hidden units in an unsupervised fashion. However, the performance of LHUC is low when fast adaptation is performed with a small amount of data. In addition, the per-speaker footprint of LHUC is also considerably huge. The proposed subspace LHUC aims to address these issues. In subspace LHUC, condition dependent parameters are estimated in a subspace and connected to various layers through a new set of weights that are adaptively trained. First, we initialize the condition dependent subspace from the i-vector during training and then during test-time adaptation, a shift is estimated for each test condition. we have evaluated the proposed method in two benchmark ASR tasks: the Aurora4 [36] and the Augmented Multi-party Interaction (AMI) [37] individual headset microphone (IHM) tasks.

The rest of the paper is organized as follows. Section 2 describes the proposed subspace LHUC method while in Section 3 the test-time adaptation is discussed. In Section 4 we give the details of our experimental setup. The results are reported in Section 5 and we conclude our work in Section 6.

2. Subspace LHUC

A DNN hidden layer learns a more abstract representation ($\mathbf{h}^l$) from the input to that layer ($\mathbf{h}^{l-1}$) and the output layer classifies the targets using a softmax function.

$$\mathbf{h}^l = \sigma(\mathbf{W}^l \mathbf{h}^{l-1} + \mathbf{b}^l) \quad (1)$$

where σ is the sigmoid activation function, $\mathbf{W}^l$ is the weight matrix for layer l, and $\mathbf{b}^l$ is the bias vector for layer l, respectively.

The adaptation methods that employ a SD feature transformation on the W^l like LIN [18], LHN [23], LON [19] can be

Copyright © 2016 ISCA

http://dx.doi.org/10.21437/Interspeech.2016-1249

represented as follows:

$$\mathbf{h}_s^l = \sigma(\mathbf{W}^l \mathbf{A}_s^l \mathbf{h}^{l-1} + \mathbf{b}^l) \quad (2)$$

where $\mathbf{A}_s^l$ is the SD transformation matrix for layer l.

However, in these adaptation methods, estimating the full matrix $\mathbf{A}_s^l$ usually introduce millions of SD parameters. Therefore, it is possible to reduce the per-speaker footprint by estimating the diagonal elements of $\mathbf{A}_s^l$.

$$\mathbf{A}_s^l = diag(\mathbf{a}_s^l) \quad (3)$$

In LHUC adaptation, an additional constraint is applied to the diagonal elements which restrict them in the range of $[0, 2]$ as given in equation 4.

$$\mathbf{a}_s^l = 2 \times \sigma(\mathbf{r}_s^l) \quad (4)$$

where $\mathbf{r}_s^l$ is the SD parameter vector for speaker s of layer l and these $\mathbf{r}_s^l$ parameter values are estimated for the test speaker using the adaptation data.

In the subspace LHUC method, we propose to learn $\mathbf{r}_s^l$ using a low-dimensional speaker representation as given below:

$$\mathbf{r}_s^l = \mathbf{U}^l \mathbf{v}_s \quad (5)$$

where $\mathbf{v}_s$ is a low dimensional vector for speaker s, and $\mathbf{U}^l$ is the connecting weight matrix for layer l and is learned using the training data. Furthermore, this method reduces the per-speaker footprint considerably as $|\mathbf{v}_s| \ll |\mathbf{r}_s^l|$. In this paper, we use the i-vector as the low dimensional representation, however, it is possible to use other representations like bottleneck vectors.

The $\mathbf{v}_s$ is estimated independent of the DNN training, therefore adding a nonlinear layer specific to $\mathbf{v}_s$ enables to learn a more abstract representation during DNN training. In addition, it allows to learn a representation that is more suitable for scaling the hidden units of the original model.

$$\hat{\mathbf{v}}_s = \sigma(\mathbf{\Gamma} \mathbf{v}_s), \quad (6)$$

$$\mathbf{r}_s^l = \sigma(\mathbf{U}^l \hat{\mathbf{v}}_s) \quad (7)$$

where $\mathbf{\Gamma}$ is the transformation matrix for the repesentation $\mathbf{v}_s$.

3. Test-time Adaptation

In our previous work [7], we showed that it is possible to combine adaptative training with test-time discriminative adaptation methods to improve the performance. Based on that result we propose to use unsupervised adaptation with subspace LHUC. In our method, the adaptation can be conducted in one of the two ways as mentioned below. We describe these ways of adaptation to the model with the i-vector specific nonlinear layer. The adaptation of the model with direct connections to the i-vector is similar.

3.1. Shift Adaptation

In shift adaptation, given the adaptation data from speaker s, we estimate a shift δ_s^l for all the layers as given in equation 8.

$$\mathbf{r}_s^l = \mathbf{U}^l(\hat{\mathbf{v}}_s + \delta_s^l) \quad (8)$$

Table 1: The number of adaptation parameters for each technique. $|\mathbf{v}_s|$ is the dimensionality of the i-vector and the $|\hat{\mathbf{v}}_s|$ is the dimensionality of the i-vector specific hidden layer.

Adaptation Technique	Number of Speaker Parameters
Shift	$\|\mathbf{v}_s\| + l \times \|\hat{\mathbf{v}}_s\|$
Constrained Shift	$\|\mathbf{v}_s\| + \|\hat{\mathbf{v}}_s\|$

3.2. Constrained Shift Adaptation

In constrained shift adaptation, this shift for speaker s is shared among all the layers as given in equation 9.

$$\mathbf{r}_s^l = \mathbf{U}^l(\hat{\mathbf{v}}_s + \delta_s) \quad (9)$$

The choice of the adaptation technique depends on factors like the quality of hypotheses, per-speaker footprint requirements and the amount of available adaptation data. Therefore, we have summarized the number of adaptation parameters for the two techniques in Table 1. We evaluate each technique in section 5.

We summarize the major steps in the subspace LHUC method below.

1) Train the initial DNN model. ($\mathbf{W}^l$ and $\mathbf{b}^l$ for)

2) Using i-vectors ($\mathbf{v}_s$) for training speakers, learn $\mathbf{U}^l, \mathbf{\Gamma}$ while keeping initial model weights fixed.

3) Extract the i-vectors for testing speakers.

4) Perform shift (δ_s^l) or constrained shift (δ_s) adaptation.

5) Perform final decoding.

4. Experimental Setup

4.1. Aurora4

We use the Aurora4 multi-condition training set with 83 speakers for training and the development set with 10 speakers for validation. The results are reported on the test set with 8 speakers.

First, we extracted the MFCC features from the speech using a 25-ms window and a 10-ms frame-shift. We obtain the LDA features by first splicing 7 frames of 13-dimensional MFCCs and then projecting downwards to 40 dimensions using LDA. A single semi-tied covariance (STC) transformation [38] is applied on top of the LDA features. The GMM-HMM system for generating the alignments for DNN training is trained on top of these 40 dimensional LDA+STC features.

The DNN-HMM baseline is trained on the LDA+STC features that span a context of 11 neighboring frames. Before being presented to the DNN, cepstral mean variance normalization (CMVN) is performed on the features globally. To train the network, layer-wise discriminative pre-training is used. The initial DNN has 7 sigmoid hidden layers with 2048 units per layer, and 2031 Senones as the outputs. All the DNNs are trained to optimize the cross-entropy criterion with a mini-batch size of 256. We use computational network toolkit (CNTK) [39] to train the DNNs. The Kaldi toolkit [40] is used to built the GMM-HMM systems and for the i-vector extraction. The i-vectors are trained on top of the same 40 dimensional LDA+STC features. The UBM consist of 128 Gaussians. We extracted i-vectors that are of 100 dimensions. For speaker-level experiments, we used the

Table 2: *Word Error Rates (WER %) for Models trained on LDA features*

Model	Eval Set	#Speaker Params
Baseline	11.9	-
+ LHUC	10.0	2048*7 = 14336

speaker i-vector and for utterance-level experiments, we used utterance-level i-vectors. We use all the test speaker data for the speaker i-vector extraction. All the decodings are performed with the pruned 5K trigram language model of WSJ0.

4.2. AMI IHM

For the next set of experiments, we used the AMI IHM corpus which contains about 100 hours of meetings conducted in English. We use the ASR split [41] of the corpus where the 78 hours of the data is used for training while about 9 hours each is used for evaluation and development sets. We use 90% of the training set for training and the rest is used as the validation set. The results are reported in the evaluation set.

For AMI experiments, we follow exactly the same steps mentioned in the Aurora4 experimental setup to generate the LDA+STC features. The DNN baseline is trained on the same LDA+STC features and has 6 sigmoid hidden layers with 2048 units per layer, and around 4000 Senones as the outputs. All the DNNs are trained to optimize the cross-entropy criterion with a mini-batch size of 256. As in Aurora4 experiments, we use CNTK to train the DNNs and Kaldi toolkit to train GMM-HMM systems as well as for i-vector extraction. For decodings, we use the trigram language model as used in Kaldi which is an interpolation of trigram language models trained on AMI and Fisher English transcripts.

5. Results

5.1. Aurora4 Results

The Table 2 reports the performance of the baseline system (11.9) and the improvement that can be achieved by performing LHUC per-speaker. As it can be clearly seen, LHUC improves the performance significantly (from 11.9 to 10.0). It is worth highlighting that this improvement is similar to the improvements reported in the literature.

We present our speaker-level subspace LHUC adaptation results in Table 3. Both subspace LHUC with direct connections to the speaker i-vector (M_1) and subspace LHUC with the i-vector specific nonlinear layer (M_2) approaches report similar improvements over the baseline system when none of the adaptation techniques are employed. However, M_1 and M_2 approaches performs differently with (test-time) adaptation. As it can be clearly seen, when shift adaptation is used M_2 (9.7) performs considerably better than M_1 (10.0). The trend is different for constrained shift adaptation where the performance of M_1 (9.8) is better than the that of M_2 (9.9). This behavior can be explained as follows. The shift adaptation is more powerful than the constrained shift adaptation due to the higher number of speaker parameters. However, the condition-specific layer allows the M_2 to learn a more robust representation that is suitable for adaptation. Therefore, when M_2 is used with shift adaptation, it performs better and constrained adaptation limits the adaptation power due to the small number of parameters. In contrast, M_1 is more sensitive to the adaptation changes than M_2. Therefore, it performs better with the constrained shift adaptation where the number of adaptation parameters is small. Moreover, it is worth highlighting both subspace LHUC models perform better than the standard LHUC. Furthermore, subspace LHUC reduces the per-speaker footprint by 94.4% and 98.6% with shift and constrained shift adaptation respectively.

Table 3: *The performance comparison of speaker-level subspace LHUC adaptation. In M_1, the i-vectors are directly connected whereas in M_2, i-vector specific nonlinear layer is used.*

Adaptation Technique	M_1	M_2	#Speaker Params
None	11.0	11.0	100
shift	10.0	9.7	800
constrained shift	9.8	9.9	200

Table 4: *WER % for Models trained on LDA features with utterance-level LHUC adaptation*

Model	Eval Set
Baseline	11.9
+ utterance-level LHUC	11.5

All the above results are about performing speaker-level adaptation. We are interested to see how these methods perform when a small amount of adaptation data is available. To investigate this, we conducted utterance-level adaptation. The utterance-level adaptation with standard LHUC is reported in Table 4. As expected, the performance improvement (from 11.9 to 11.5) for utterance-level adaptation is considerably smaller for standard LHUC.

In Table 5, we present the results of the utterance-level experiments for subspace LHUC. Even without any adaptation, both M_1 (10.8) and M_2 (10.8) perform significantly better than the standard LHUC (11.5). In addition, when test-time adaptation is used, performance improves significantly with the best value of 10.2 for M_2 with constrained shift adaptation. For both M_1 and M_2, the constrained shift adaptation perform significantly better than the shift adaptation. This is clearly due to the smaller number of parameters in constrained shift adaptation that allows robust adaptation with smaller amounts of adaptation data.

Since constrained shift adaptation outperforms shift adaptation for utterance-level experiments of subspace LHUC, it is worthwhile to investigate the utterance-level performances for smaller subspace dimensions. In Figure 1 we investigate the effect of the subspace dimensionality for utterance-level subspace LHUC for the model with i-vector specific nonlinear layer (M_2). As it can be clearly seen, the performance improved considerably when the subspace dimensionality is increased from 10 to 25 and 25 to 50 for both i-vector initialized and con-

Table 5: *The performance comparison of utterance-level subspace LHUC adaptation*

Adaptation Technique	M_1	M_2
None	10.8	10.8
shift	10.6	10.5
constrained shift	10.4	10.2

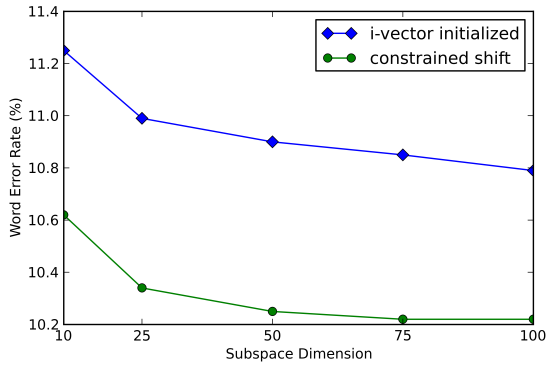

Figure 1: WER (%) for various subspace dimensions ($|\hat{\mathbf{v}}_s|$) of subspace LHUC for utterance-level adaptation.

strained shift adapted systems. However, for subspace dimensionalities of 50, 75, 100 the performances are similar.

5.1.1. Histogram Analysis

In this section, we analyze the amplitude parameters of the LHUC adaptation for both speaker-level and utterance-level adaptation experiments. The first and second columns of Figure 2 show the histograms for speaker-level and utterance-level experiments respectively. For speaker-level adaptation, the histogram shapes for both standard LHUC and subspace LHUC after adaptation is similar. In contrast, with utterance-level adaptation, standard LHUC amplitude values hardly changes (note that the x and y scales are different for utterance-level standard LHUC plot from other plots). However, both speaker-level and utterance-level plots for subspace LHUC before adaptation are similar. This is because $\mathbf{U}^l$, $\mathbf{\Gamma}$ is learned from the training data and utterance-level i-vector is extracted independently of the DNN. Therefore, subspace LHUC provides a better initialization for adaptation.

5.2. AMI Results

In the next set of experiments, we perform speaker-level and utterance-level adaptation on AMI IHM dataset. For AMI, we only used the subspace LHUC model with the i-vector specific hidden layer (M_2). Furthermore, we use $|\hat{\mathbf{v}}_s|$ = 50 for all AMI IHM subspace LHUC models.

Table 6 shows the results for speaker-level adaptation. Both standard LHUC and subspace LHUC improved the performance significantly. Similar to the Aurora4 experiments, subspace LHUC with shift adaptation reported the best performance (25.9 %). In addition, when subspace LHUC is used with shift adaptation the per-speaker footprint reduces by 96.7 % over the standard LHUC adaptation. The per-speaker footprint reduction is 98.8% for subspace LHUC with constrained shift adaptation.

Table 7 reports the results for the utterance-level adaptation on AMI IHM corpus. As shown in the table, standard LHUC reported no improvements when utterance-level adaptation is performed. However, subspace LHUC improved the performance for both before and after the adaptation. A relative improvement of 3.4% is reported with subspace LHUC before the adaptation. The performance improved further with constrained adaptation to 4.5% relatively.

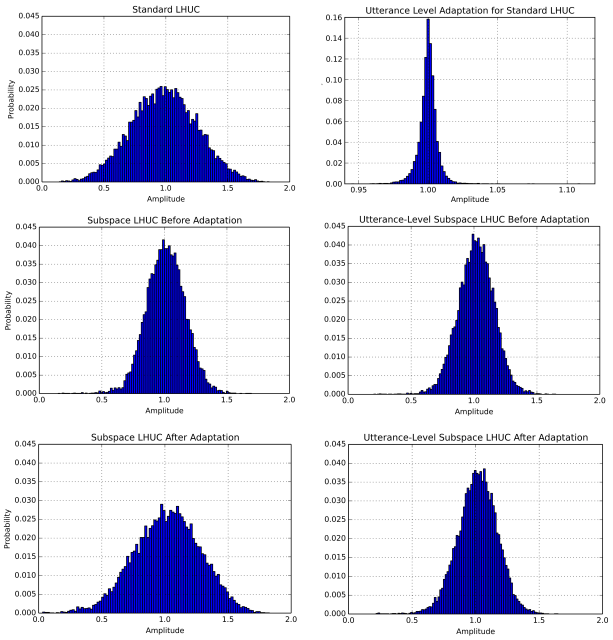

Figure 2: Histrograms for the amplitude values of standard LHUC, subspace LHUC before adaptation and subspace LHUC after adaptation for both speaker-level and utterance-level adaptation.

Table 6: *Eval Set WER % for Models trained on AMI IHM task for speaker-level experiments*

Model	WER	#Speaker Params
Baseline	29.0	-
+speaker-level LHUC	26.1	12288
M_2	27.7	100
+ constrained shift	26.2	150
+ shift	25.9	400

6. Conclusions

In this work, we have proposed the subspace LHUC, where the SD parameters are estimated in a subspace and connected to various layers through a new set of adaptively trained weights. We evaluated the subspace LHUC in the Aurora4 and the AMI IHM tasks. Experimental results showed that the subspace LHUC outperforms standard LHUC adaptation. With utterance-level fast adaptation, subspace LHUC achieved 11.3% and 4.5% relative improvements over the standard LHUC for the Aurora4 and AMI IHM tasks respectively. Furthermore, the subspace LHUC reduced the per-speaker footprint at least by 94% over the standard LHUC adaptation.

Table 7: *Eval Set WER % for Models trained on AMI IHM task for utterance-level experiments*

Model	WER
Baseline	29.0
+utterance-level LHUC	29.0
M_2	28.0
+ constrained shift	27.7

7. References

[1] J. Gauvain and C. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[2] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[3] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *ICSLP*, vol. 2. ISCA, 1996, pp. 1137–1140.

[5] M. Gales, "Cluster adaptive training of hidden markov models," vol. 8, no. 4, pp. 417–428, 2000.

[6] L. Samarakoon and K. Sim, "Learning factorized transforms for speaker normalization," in *ASRU*. IEEE, 2015.

[7] ——, "On combining i-vectors and discriminative adaptation methods for unsupervised speaker normalization in DNN acoustic models," in *ICASSP*. IEEE, 2016.

[8] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP*. IEEE, 2013, pp. 7893–7897.

[9] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *ICASSP*. IEEE, 2014, pp. 6334–6338.

[10] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

[11] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *ICASSP*. IEEE, 2013, pp. 7942–7946.

[12] D. Yu, , and L. Deng, *Automatic Speech Recognition - A Deep Learning Approach*. New York: Springer London, 2015.

[13] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*. IEEE, 2000, pp. 1635–1638.

[14] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *ICASSP*. IEEE, 2013, pp. 6975–6979.

[15] Y. Liu, P. Zhang, and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," in *ICASSP*. IEEE, 2014, pp. 5542–5546.

[16] S. Liu and K. Sim, "On combining DNN and GMM with unsupervised speaker adaptation for robust automatic speech recognition," in *ICASSP*. IEEE, 2014, pp. 195–199.

[17] ——, "Temporally varying weight regression: A semi-parametric trajectory model for automatic speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 1, pp. 151–160, 2014.

[18] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Eurospeech*. ISCA, 1995, pp. 2183–2186.

[19] B. Li and K. Sim, "Comparison of discriminative input and output transformation for speaker adaptation in the hybrid NN/HMM systems," in *INTERSPEECH*. ISCA, 2010, pp. 526–529.

[20] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker-adaptation for hybrid HMM-ANN continuous speech recognition system," 1995.

[21] J. Trmal, J. Zelinka, and L. Müller, "Adaptation of a feedforward artificial neural network using a linear transform," in *Text, Speech and Dialogue*. Springer, 2010, pp. 423–430.

[22] Y. Xiao, Z. Zhang, S. Cai, J. Pan, and Y. Yan, "A initial attempt on task-specific adaptation for deep neural network-based large vocabulary continuous speech recognition," in *INTERSPEECH*. ISCA, 2012.

[23] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," in *ICASSP*. IEEE, 2006, pp. 1189–1192.

[24] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*. IEEE, 2011, pp. 24–29.

[25] S. Xue, H. Jiang, and L. Dai, "Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition," in *ISCSLP*. IEEE, 2014, pp. 1–5.

[26] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *ICASSP*. IEEE, 2014, pp. 6359–6363.

[27] K. Kumar, C. Liu, K. Yao, and Y. Gong, "Intermediate-layer DNN adaptation for offline and session-based iterative speaker adaptation," in *INTERSPEECH*. ISCA, 2015.

[28] S. Dupont and L. Cheboub, "Fast speaker adaptation of artificial neural networks for automatic speech recognition," in *ICASSP*. IEEE, 2000, pp. 1795–1798.

[29] J. Stadermann and G. Rigoll, "Two-stage speaker adaptation of hybrid tied-posterior acoustic models," in *ICASSP*. IEEE, 2005, pp. 977–980.

[30] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *ICASSP*. IEEE, 2013, pp. 7947–7951.

[31] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*. IEEE, 2014, pp. 171–176.

[32] Y. Huang and Y. Gong, "Regularized sequence-level deep neural network model adaptation," in *INTERSPEECH*. ISCA, 2015.

[33] T. Tian, Q. Yanmin, Y. Maofan, Z. Yimeng, and K. Yu, "Cluster adaptive training for deep neural network," in *ICASSP*. IEEE, 2015, pp. 4325–4329.

[34] C. Wu and M. J. F. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *ICASSP*. IEEE, 2015, pp. 4315–4319.

[35] L. Lee and R. Rose, "Speaker normalization using efficient frequency warping procedures," in *ICASSP*, vol. 1. IEEE, 1996, pp. 353–356.

[36] N. Parihar, J. Picone, D. Pearce, and H. Hirsch, "Performance analysis of the aurora large vocabulary baseline system," in *EUSIPCO*. IEEE, 2004, pp. 553–556.

[37] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The ami meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.

[38] M. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[39] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang *et al.*, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR, Microsoft Research, 2014, http://codebox/cntk, Tech. Rep., 2014.

[40] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *ASRU*. IEEE, 2011.

[41] P. Swietojanski, A. Ghoshal, and S. Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *ASRU*. IEEE, 2013, pp. 285–290.