FEATURE COMPENSATION USING LINEAR COMBINATION OF SPEAKER AND ENVIRONMENT DEPENDENT CORRECTION VECTORS

Xiong Xiao[1], *Jinyu Li*[2], *Eng Siong Chng*[1,3], *Haizhou Li*[1,3,4]

[1]Temasek Lab@NTU, Nanyang Technological University, Singapore
[2]Microsoft Corporation, USA
[3]School of Computer Engineering, Nanyang Technological University, Singapore
[4]Department of Human Language Technology, Institute for Infocomm Research, Singapore

xiaoxiong@ntu.edu.sg, jinyli@microsoft.com, aseschng@ntu.edu.sg, hli@i2r.a-star.edu.sg

ABSTRACT

In this paper, we study a novel way to compensate speech features to counter the effects of speaker variations and environment distortions in speech recognition. For each homogeneous cluster of speech data, e.g. a specific speaker and environment combination, a set of correction vectors are learnt. A correction vector measures the deviation of features in a small region of feature space due to the speaker and environment effects. From a heterogenous training set, dozens of sets of correction vectors are learnt, each from a homogenous subset of the data. During testing, those correction vector sets are linearly combined to compensate test feature vectors. The combination weights are estimated by maximizing the likelihood (ML) of the compensated features with respect to a reference model, which is a simplified version of the acoustic model used for speech recognition. In addition, variance compensation is applied to condition the variances of the compensated features during weight estimation. Experimental results on Aurora-4 multi-condition training task show that the proposed correction vector combination method reduces the word error rate (WER) to 14.97% from mean and variance normalization baseline (16.32%) for noisy test sets 2-7. In addition, the proposed ML weight estimation consistently outperforms the posterior weights used in previous studies, such as multi-environment SPLICE.

Index Terms— feature compensation, environment combination, correction vectors, Aurora-4, robust speech recognition.

1. INTRODUCTION

Speech recognition performance degrades significantly when the test acoustic condition is different from the training acoustic condition [1]. The mismatch could be due to many factors, e.g. speaker characteristics, background noises, and transmission channels. Generally speaking, we can improve the speech recognition performance by either compensating/normalizing the features [2–7] or adapting the acoustic model [5, 8–10], or both. In addition, these two approaches could be used together with multi-condition training, i.e. use speech data collected from various acoustic conditions to train the acoustic model, to further improve the system robustness [11, 12].

Reference speaker weighting (RSW) [13, 14] and eigenvoice [15] are fast model adaptation methods that are effective even with several seconds of adaptation data. In RSW, a set of speaker-dependent acoustic models are trained. Only the mean vectors are updated and the covariance matrices are shared among all speaker models. From each speaker model, a mean supervector is constructed by concatenating all the mean vectors. In testing phase, the mean supervectors are linearly combined to form the mean supervector of the adapted acoustic model using maximum likelihood (ML) criterion. As the parameter of RSW is the weights of speaker dependent supervectors, which is usually less than 100, very few data is required for RSW to be effective. Recently, this concept is extended to also performing simultaneous environment selection by using an L1 norm regularizer [16]. Eigenvoice [15] is similar to RSW, except that a small number of principal components are learnt from the speaker-dependent supervectors and used during test.

Compared to model adaptation, processing features has some advantages. For example, we usually do not need to use the complex acoustic model for feature compensation and there is no need to do multi-pass decoding. More importantly, the adapted features can be used with any acoustic models, e.g. deep neural network (DNN) based acoustic model [17].

In this paper, we apply the concept of RSW to feature compensation. The basic idea is to compensate test feature vectors using linear combination of correction vector sets, each set representing feature distortion in a specific speaker and environment combination. This is conceptually similar to, but fundamentally different from RSW, in which the acoustic models are adapted. We use the ML criterion to estimating the linear combination weights, with a variance compensation term to prevent the variances of the compensated features from shrinking significantly. The proposed method has some common characteristics with the popular SPLICE feature compensation method [4], e.g. both using region-dependent correction vectors, but without the need of stereo training data. We will discuss the differences between the two in details in the paper.

The rest of this paper is organized as follows. In section 2, we will describe the proposed feature compensation method, and also discuss its relationship to RSW and SPLICE. In section 3, the proposed method is evaluated on the Aurora-4 multi-condition training task. Finally, we will conclude in section 4.

2. FEATURE COMPENSATION WITH PRETRAINED CORRECTION VECTORS

Suppose we have a heterogenous training set which contains speech data recorded from various acoustic conditions. One question is how to make use of this training set to compensate test features such that the mismatch between the acoustic model, either trained from the heterogenous training set or a clean training set, and the noisy test set are reduced. In this study, we study the approach of using sets of pretrained correction vectors, each trained from a homogenous acoustic condition, to compensate the noisy features.

2.1. Characterizing Feature Distortions by Region-Dependent Correction Vectors

For a homogenous acoustic condition, e.g. a specific speaker and noise combination, it is reasonable to assume that for a small region in the feature space, the effects of speaker and noise can be approximated by a shift of feature vectors in the region. Such concept is widely used in previous studies. For example, in SPLICE [4], a region dependent correction vector is usually trained with stereo data to approximate the distortions caused by noise. In model adaptation methods [8], among all the model parameters, it is the most important to adapt the mean vectors of the Gaussians, in which the difference between the adapted and original mean vectors can be considered as a shift of feature space. From another point of view, region dependent correction vectors in cepstral domain is similar to filtering of speech spectrum using a set of predefined filters in frequency domain, where the filter's frequency response is determined by the correction vectors. Therefore, simply using correction vectors can be very powerful, if the feature space is divided into sufficiently large number of regions and the region selection is accurate.

There are several ways to obtain the region-dependent correction vectors. One way is to use stereo data which can be obtained by adding recorded noise to clean speech, or by simultaneously recording speech using both close-talk and far-talk microphones. Either ML or minimum mean square error (MMSE) criterion can be used to find correction vectors from stereo data [18]. A limitation of this approach is that stereo data is not always available, e.g. in the broadcast news transcription task.

The correction vectors used in this study are obtained by using model adaptation method. First, we train a universal GMM from a heterogenous training set, in which each Gaussian represents a region of the feature space. Then we adapt the universal GMM to each homogenous subset of the training set. The adaptation could be done by conventional model adaptation methods, such as MLLR [8] or MAP [9] mean adaptation. After that, the correction vector for the m^{th} region of the i^{th} acoustic condition is obtained by

$$\mathbf{r}_i^{(m)} = \boldsymbol{\mu}^{(i,m)} - \boldsymbol{\mu}^{(m)} \tag{1}$$

where $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\mu}^{(i,m)}$ are the mean vectors of the m^{th} Gaussian in the universal and i^{th} adapted GMM's, respectively. The correction vector represents the average shift of feature vectors in a region due to different acoustic conditions in the universal and adapted GMM's.

The advantage of using model adaptation method to find correction vectors is that there is no requirement to have stereo data. The only requirement is to cluster the heterogenous training data into homogenous subsets, which can be achieved with plenty of methods.

2.2. Feature Compensation by Combined Correction Vectors

For a homogenous acoustic condition, we use a correction vector to compensate the feature vectors in a small region in feature space:

$$\hat{\mathbf{o}}_{ti} = \mathbf{o}_t - \mathbf{r}_i^{(m)} \tag{2}$$

where $\mathbf{o}_t$ is the noisy feature vector, and $\hat{\mathbf{o}}_{ti}$ is the compensated feature vector by using the correction vectors from the i^{th} environment. In (2), we assume that a feature vector is solely belonging to the m^{th} region. However, in practice, such hard mapping of feature vectors to regions is not optimal. We can use the universal GMM to partition the feature space into M regions and use soft-decision mapping:

$$\hat{\mathbf{o}}_{ti} = \mathbf{o}_t - \sum_{m=1}^{M} P(m|\mathbf{o}_t)\mathbf{r}_i^{(m)} = \mathbf{o}_t - \mathbf{r}_{ti} \tag{3}$$

$$P(m|\mathbf{o}_t) = \frac{c_m \mathcal{N}(\mathbf{o}_t, \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)})}{\sum_{m'=1}^{M} c_{m'} \mathcal{N}(\mathbf{o}_t, \boldsymbol{\mu}^{(m')}, \boldsymbol{\Sigma}^{(m')})} \tag{4}$$

where $P(m|\mathbf{o}_t)$ is the posterior probability of region m after observing the noisy feature $\mathbf{o}_t$. c_m, $\mu^{(m)}$, and $\Sigma^{(m)}$ are the prior weight, mean vector, and covariance matrix of the m^{th} Gaussian in the GMM, respectively. In (3), $\mathbf{r}_{ti} = \sum_{m=1}^{N} P(m|\mathbf{o}_t)\mathbf{r}_i^{(m)}$ is the overall correction vector from environment i and is time dependent.

In test phase, assume that we have correction vectors for I acoustic conditions. Given a test utterance without speaker or environment information, we can choose a condition that is the closest to the test utterance in some sense and use its correction vectors to compensate the test features. However, as it is hard to guarantee that the test utterance matches exactly a single training condition, it's better to combine correction vectors from different conditions:

$$\hat{\mathbf{o}}_t = \mathbf{o}_t - \sum_{i=1}^{I} w_i \mathbf{r}_{ti} = \mathbf{o}_t - \mathbf{R}_t \mathbf{w} \tag{5}$$

where $\hat{\mathbf{o}}_t$ is the final compensated feature vector for frame t and w_i is the weight of correction vectors from the i^{th} environment. $\mathbf{R}_t = [\mathbf{r}_{t1}, ..., \mathbf{r}_{tI}]$ is the matrix of correction vectors of all environments for time t, and $\mathbf{w} = [w_1, ..., w_I]$ is the vector of weights.

The feature compensation method shown in (5) is characterized by a time dependent correction matrix $\mathbf{R}_t$ which is built from I environment dependent corrector vector sets via the posterior probabilities of the regions and a time-independent weight vector $\mathbf{w}$. As the number of environments is relatively small, e.g. 100, the weights could be reliably estimated even from a short utterance.

2.3. Estimation of Linear Combination Weights

There are several ways to find the linear combination weights in (5). One possible way is to set the weights of the i^{th} set to its posterior probability given the observed features:

$$\hat{w}_i = P(i|\mathbf{o}_t) = \frac{p(\mathbf{o}_t|i)P(i)}{\sum_{i=1}^{I} p(\mathbf{o}_t|i)P(i)} \tag{6}$$

where $P(i)$ is the prior probability of acoustic condition i which is usually ignored, and $p(\mathbf{o}_t|i)$ is the likelihood of features in condition i and could be represented by the acoustic condition dependent GMM. The posterior in (6) is noisy and could be smoothed along time or averaged over the whole test utterance. We call this way of weight estimation the posterior method [19].

Although using posteriors of environments as weights are intuitive for combining the correction vector sets, it may not produce compensated features that fit the acoustic model. To address this issue, we propose to find the linear combination weights by maximizing the likelihood of the compensated features w.r.t. the speech recognition system's acoustic model. As the acoustic model usually contains a large number of Gaussians, we can also maximize the likelihood on a simpler reference model for less computational cost, e.g. a GMM that is trained from the same training data as the acoustic model. In this study, we assume the acoustic model and the region-partitioning universal GMM are trained from the same heterogenous training set, hence, it is reasonable to use the universal

GMM as the reference GMM. Then the weights will be obtained by solving the following problem:

$$\hat{\mathbf{w}} \;=\; \arg\max_{\mathbf{w}} \sum_{t=1}^{T} \text{log} p(\hat{\mathbf{o}}_t|\Lambda) - \frac{\alpha}{2}||\mathbf{w}||^2 \qquad (7)$$

where $\Lambda = \{c_m, \mu^{(m)}, \Sigma^{(m)}|m = 1, ..., M\}$ is the parameters of the reference/universal GMM. Note that there is also an L2 norm regularization term in the objective function for robust estimation of weights, especially when the test utterance is short. The parameter α is used to control how much regularization we want to impose and is empirically tuned.

To solve the weight estimation problem, we can use the Expectation Maximization (EM) algorithm [20]. The auxiliary function is

$$\begin{aligned} Q &= \sum_{t=1}^{T}\sum_{m=1}^{M} \gamma_m(t) \text{log}\mathcal{N}(\mathbf{o}_t - \mathbf{R}_t\mathbf{w}; \boldsymbol{\mu}^{(m)}, \boldsymbol{\Sigma}^{(m)}) - \frac{\alpha}{2}||\mathbf{w}||^2 \\ &= -\frac{1}{2}\mathbf{w}^T\mathbf{G}\mathbf{w} + \mathbf{w}^T\mathbf{p} - \frac{\alpha}{2}||\mathbf{w}||^2 + K \qquad (8) \end{aligned}$$

where

$$\mathbf{G} \;=\; \sum_{mt} \gamma_m(t)\mathbf{R}_t^T(\boldsymbol{\Sigma}^{(m)})^{-1}\mathbf{R}_t \qquad (9)$$

$$\mathbf{p} \;=\; \sum_{mt} \gamma_m(t)\mathbf{R}_t^T(\boldsymbol{\Sigma}^{(m)})^{-1}(\mathbf{o}_t - \boldsymbol{\mu}^{(m)}) \qquad (10)$$

where $\gamma_m(t) = p(m|\mathbf{o}_t)$ and K is a constant not dependent on $\mathbf{w}$.

The weight estimation in (8) is a ridge regression problem [21] and the closed form solution can be easily found by

$$\hat{\mathbf{w}} \;=\; (\mathbf{G} + \alpha\mathbf{I})^{-1}\mathbf{p} \qquad (11)$$

where $\mathbf{I}$ is the identity matrix.

2.4. Variances Compensation for ML Estimation

Although the solution in (11) can improve the match between the compensated features and the reference model, it tends to reduce the variance of the compensated features. In our preliminary experiments, we found that the shrinking variance results in significantly more deletion errors and high overall WER. This is because the solution in (11) tries to map the feature vectors to the mean vectors of the reference model. Similar problem is also observed in voice conversion [22] and feature normalization [6, 23]. To address this problem, we introduce a term to encourage the increasing of log determinant of the covariance matrix of the compensated features as follows:

$$Q2 \;=\; -\frac{1}{2}\mathbf{w}^T\mathbf{G}\mathbf{w} + \mathbf{w}^T\mathbf{p} - \frac{\alpha}{2}||\mathbf{w}||^2 + \frac{\beta}{2}\log|\hat{\mathbf{C}}| \qquad (12)$$

where $\hat{\mathbf{C}} = \frac{1}{T}\sum_{t=1}^{T}(\hat{\mathbf{o}}_t - \hat{\boldsymbol{\mu}})(\hat{\mathbf{o}}_t - \hat{\boldsymbol{\mu}})^T$ and $\hat{\boldsymbol{\mu}} = \frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{o}}_t$ are the sample covariance matrix and sample mean of the compensated feature vectors estimated over the current test utterance. Note that the variance compensation in (12) is the same as the Jacobian compensation for vocal tract length normalization (VTLN) [24]. As only offset vectors are used and no linear transform is used in the proposed feature compensation, there is no well-defined Jacobian. Hence, the weight of variance compensation is not necessarily equal to the theoretical value of 1 and will be empirically determined.

The maximization of (12) is complicated, so we use gradient-based method such as L-BFGS [25] to find the solution of the weights iteratively. Due to page limitation, we skip the derivation of the gradients and directly present the gradients as follows.

$$\frac{\partial Q2}{\partial \mathbf{w}} \;=\; -\mathbf{G}\mathbf{w} + \mathbf{p} - \alpha\mathbf{w} + \frac{\beta}{2}\frac{\partial \log|\hat{\mathbf{C}}|}{\partial \mathbf{w}} \qquad (13)$$

$$\frac{\partial \log|\hat{\mathbf{C}}|}{\partial w_i} \;=\; \text{Tr}\Big(\hat{\mathbf{C}}^{-1}(\mathbf{A} + \mathbf{A}^T)\Big) \qquad (14)$$

where $\mathbf{A} = \sum_{t=1}^{T}(\hat{\mathbf{o}}_t - \hat{\boldsymbol{\mu}})\tilde{\mathbf{r}}_{t,i}^T$. $\text{Tr}(\cdot)$ is the trace of a matrix. $\tilde{\mathbf{r}}_{t,i}$ is the i^{th} column of matrix $\tilde{\mathbf{R}}_t = \bar{\mathbf{R}} - \mathbf{R}_t$, where $\bar{\mathbf{R}} = \frac{1}{T}\sum_t \mathbf{R}_t$.

2.5. Relationship to RSW and SPLICE

The proposed feature compensation method is motivated by the RSW model adaptation method. In RSW, the acoustic condition-dependent mean supervectors are linearly combined to adapt the mean of the acoustic model to fit the test data. In the proposed method, the acoustic condition dependent correction supervectors are linearly combined to compensate the features to match the original acoustic model. From (1), it can be easily seen that the acoustic condition dependent correction supervector is just the difference between the reference model's mean supervector and the condition dependent mean supervector. Despite this relationship, our proposed feature compensation method is different from RSW model adaptation, which is obvious when we compare the auxiliary function in (12) and that of RSW. In addition, our feature compensation method has frame-dependent correction vectors as shown in (5), while RSW has Gaussian dependent correction vectors.

The proposed feature compensation method shares some common spirit with the SPLICE method. For example, both use region dependent correction vectors and acoustic condition dependent corrections. However, there are several difference between these two methods. First, we used ML criterion to estimate the linear combination weights to ensure that the compensated features will match the acoustic model well. Second, we introduced the variance compensation term to prevent the compensated features from having too small variances. Third, the proposed method doesn't require stereo training data to obtain the correction vectors.

3. EXPERIMENTS

3.1. Experimental Settings

We evaluate the proposed feature compensation method on the Aurora-4 multi-condition training task [26]. The acoustic model is trained from 7138 clean and noisy utterances defined in Aurora-4. There are about 3000 tied states in the acoustic model and the emission probability of each tied state is a GMM containing 8 Gaussians.

The speech data is sampled at 16kHz sampling rate. Mel-frequency cepstral coefficients (MFCC) are used as the acoustic features. The feature vectors are 39-dimensional, containing C0 to C12 and their delta and acceleration versions. The training and testing features are both processed by utterancewise MVN [3].

The reference model used for partitioning the feature space into regions and also used as target model for ML weight estimation is built by pooling a monophone acoustic model trained from the 7138 multi-condition training utterances. There are totally 984 Gaussians in the reference model, which means that we partition the feature space into 984 regions. To obtain acoustic condition dependent correction supervectors, we adapt the monophone acoustic model to each speaker and noise combination in the training set, and obtained 581 acoustic conditions (83 speakers and 7 environment conditions).

Table 1. Recognition WER (%) on Aurora-4 task. Test case 1 is clean test, while 2-7 are 6 noisy test cases. "2-7" refers to the average WER of cases 2-7. "Posterior", "ML", and "ML+Var" are the 3 weights estimation methods described in section 2.

Methods	Test Cases							
	1	2	3	4	5	6	7	2-7
MVN	10.1	11.8	15.1	18.7	17.3	15.4	19.6	16.32
Feature Compensation								
Posterior	10.4	10.7	14.4	18.1	16.9	14.8	18.6	15.58
ML	9.6	10.6	14.0	18.5	17.0	14.0	18.4	15.42
ML+Var	9.6	10.3	13.5	17.7	16.5	13.7	18.2	14.97
Model Adaptation								
RSW	9.5	10.1	13.0	17.1	15.4	12.7	17.1	14.23

The adaptation is achieved by using 4 class-based MLLR mean transforms [8]. The speaker information is given by the corpus, while the noise type information of each training utterance is tagged manually. Note that each speaker/noise combination is not completely homogenous as the signal-to-noise ratio (SNR) of training utterance ranges from 10-20dB. Due to computational considerations, we only used 117 conditions randomly selected from the 581 training conditions in our feature compensation method. The weight of the L2 norm in (8) and (12) and the weight of the variance compensation in (12) are empirically set to 400 and 0.3, respectively. The maximum L-BFGS iterations allowed is set to 100.

For RSW, we obtain the acoustic condition dependent mean supervectors by adapting the multi-condition triphone acoustic models to each speaker/environment combination using 4 class-based MLLR mean transforms [8]. Due to computational issues, we only used 256 mean supervectors. We also used an L2 norm in the RSW to regularize the estimation of weights and found it improves performance significantly. All acoustic model training, adaptation, and speech recognition are implemented by using the HTK toolkit [27].

3.2. Performance of Feature Compensation

The performance of the proposed feature compensation method is summarized in Table 1. Three ways of estimating the environment combination weights are compared, i.e. the posterior method ("Posterior"), the ML method without variance compensation ("ML"), and ML with variance compensation ("ML+Var"). From the results, using environment posterior as weights reduces WER in all noisy test cases, but degrades the performance in clean test case. The ML estimated weights are slightly better than the posterior weights. The variance compensated ML estimation of weights performs the best in all test cases. On average, "ML+Var" achieves 1.35% absolute WER reduction over the MVN baseline.

When using environment posterior as weights, the proposed method is very similar to multi-environment SPLICE. We didn't implement the classic SPLICE as it was shown to be ineffective for Aurora-4 multi-condition training task. For example, in Table 9.3 of [18], using a 256 region SPLICE only reduces the WER of test cases 2-7 from 19.3% to 19.2%. For comparison, the posterior results in Table 1 can be roughly considered as the multi-environment SPLICE, and WER for case 2-7 is reduced from 16.32% to 15.58%. The comparison shows that using environment combination is important for medium/large vocabulary tasks like Aurora-4 when the acoustic model is already trained from noisy speech data..

We also compare with the model adaptation method RSW [13, 14]. Results in Table 1 show that RSW outperforms all feature compensation methods. There are several reasons for this observation. First, RSW uses HMM rather than GMM as reference model. Second, in RSW, two-pass decoding is used, so the Gaussian occupation probabilities used in RSW are expected to be more accurate than that used in the feature compensation methods. Third, we argue that model adaptation methods are generally more flexible than feature compensation methods. For example, in RSW, the Gaussians have new mean vectors after adaptation. This is equivalent to having a large number (≈24,000) of Guassian dependent correction vectors. However, for feature compensation, we only have 1 correction vector for each frame. Despite the better performance of RSW, feature compensation methods also enjoy some advantages. For example, there is no requirement for 2-pass decoding and the compensated features can be used with other acoustic models, such as HMM/DNN.

Table 2. Detailed error types and effect of variance compensation for ML weight estimation. Insertion penalty is set to -10 in all tests.

Methods	Error Types (%)			
	Deletion	Substitution	Insertion	Total (WER)
MVN	3.03	11.60	1.69	16.32
Feature Compensation				
Posterior	2.90	11.02	1.66	15.58
ML	**3.41**	10.67	**1.34**	15.42
ML+Var	2.95	10.53	1.49	14.97
Model Adaptation				
RSW	2.53	10.20	1.50	14.23

3.3. Effect of Variance Compensation

The results in Table 1 shows that variance compensation is important for good performance of ML-based weight estimation. In Table 2, we investigate the detailed error types of each method. It can be found that while the rest of methods do not change the insertion/deletion ratio significantly, the ML method increases the deletion error significantly, and at the same time reduces the insertion error. This is due to that the compensated features using ML-estimated weight have significantly smaller variances than the input MVN features. When variance compensation is used with ML estimation, the compensated features have roughly comparable variances as the input features. Hence, the insertion/deletion ratio also becomes more normal. The analysis shows that the variance compensation is important for the ML-based weight estimation.

4. CONCLUSIONS

In this paper, we studied a feature compensation method which linearly combines sets of pretrained acoustic condition-dependent correction vectors to optimally compensate noisy features in the ML sense. We also introduced a variance term to the objective function of the feature compensation method to prevent the compensated features from having very small variances. Results on Aurora-4 task using multi-condition trained acoustic model show that using ML criterion and variance compensation to estimate the linear weights for combining pretrained correction vectors provides better results than the previous methods.

5. REFERENCES

[1] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, “An overview of noise-robust automatic speech recognition,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, April 2014.

[2] S. Furui, “Cepstral analysis technique for automatic speaker verification,” *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[3] O. Viikki and K. Laurila, “Cepstral domain segmental feature vector normalization for noise robust speech recognition,” *Speech Communication*, vol. 25, pp. 133–147, 1998.

[4] J. Droppo, L. Deng, and A. Acero, “Evaluation of the SPLICE algorithm on the Aurora2 database,” in *Proc. InterSpeech ’01*, Aalborg, Denmark, Sept. 2001, pp. 217–220.

[5] A. Sankar and C.-H. Lee, “A maximum-likelihood approach to stochastic matching for robust speech recognition,” *IEEE Trans. Speech and Audio Processing*, vol. 4, no. 3, pp. 190–202, 1996.

[6] X. Xiao, J. Li, E. S. Chng, and H. Li, “Maximum likelihood adaptation of histogram equalization with constraint for robust speech recognition,” in *Proc. ICASSP ’11*, Prague, Czech, May 2011, pp. 5480–5483.

[7] X. Xiao, E. S. Chng, and H. Li, “Attribute-based histogram equalization (HEQ) and its adaptation for robust speech recognition,” in *Proc. InterSpeech ’13*, Lyon, France, Aug. 2013.

[8] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech and Language*, vol. 9, no. 2, pp. 171–185, Apr. 1995.

[9] J. L. Gauvain and C. H. Lee, “Maximum *a posterirori* estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[10] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, “A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions,” *Computer Speech and Language*, vol. 23, no. 3, pp. 389–405, Jul. 2009.

[11] R. Lippmann, E. Martin, and D. Paul, “Multi-style training for robust isolated-word speech recognition,” in *Proc. ICASSP ’87*, Dallas, TX, Apr. 1987, vol. 12, pp. 705–708.

[12] M. J. F. Gales, “Cluster adaptive training of hidden Markov models,” *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 4, pp. 417–428, 2000.

[13] T. J. Hazen, “A comparison of novel techniques for rapid speaker adaptation,” *Speech Communication*, vol. 31, no. 1, pp. 15 – 33, 2000.

[14] B. Mak, T.-C. Lai, and R. Hsiao, “Improving reference speaker weighting adaptation by the use of maximum-likelihood reference speakers,” in *Proc. ICASSP ’06*, 2006.

[15] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, “Rapid speaker adaptation in eigenvoice space,” *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 6, pp. 695–707, Nov 2000.

[16] X. Xiao, J. Li, E. S. Chng, and H. Li, “Lasso environment model combination for robust speech recognition,” in *Proc. ICASSP ’12*, 2012, pp. 4305–4308.

[17] F. Seide, G. Li, and D. Yu, “Conversational speech transcription using context-dependent deep neural networks.,” in *Proc. InterSpeech ’11*, 2011, pp. 437–440.

[18] J. Droppo, “Feature compensation,” in *Techniques for Noise Robust in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. Wiley, 2012.

[19] J. Droppo, A. Acero, and L. Deng, “Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system,” in *Proc. ICASSP ’01*, 2001, pp. 209–212.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[21] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, Springer, 2009.

[22] T. Toda, A. W. Black, and K. Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameters trajectory,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, Nov. 2007.

[23] X. Xiao, E. S. Chng, and H. Li, “Temporal filter design by minimum KL divergence criterion for robust speech recognition,” in *Proc. ICASSP ’13*, Vancouver, Canada, May.

[24] G. Saon, G. Zweig, B. Kingsbury, L. Mangu, and U. V. Chaudhari, “An architecture for rapid decoding of large vocabulary conversational speech,” in *Proc. InterSpeech ’03*, Geneva, Switzerland, Sept. 2003.

[25] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[26] N. Parihar and J. Picone, “Aurora working group: DSR front end LVCSR evaluation AU/384/02,” Tech. Rep., Institute for Signal and Infomation Processing, Mississippi State Univ., MS, Dec. 2002.

[27] S. Young et al., *The HTK book*, Cambridge university engineering department, 3.4 edition, 2006.