JOINT SPECTRAL AND TEMPORAL NORMALIZATION OF FEATURES FOR ROBUST RECOGNITION OF NOISY AND REVERBERATED SPEECH

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

[1]Temasek Lab@NTU, Nanyang Technological University, Singapore
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]Department of Human Language Technology, Institute for Infocomm Research, Singapore
xiaoxiong@ntu.edu.sg, aseschng@ntu.edu.sg, hli@i2r.a-star.edu.sg

ABSTRACT

In this paper, we propose a framework for joint normalization of spectral and temporal statistics of speech features for robust speech recognition. Current feature normalization approaches normalize the spectral and temporal aspects of feature statistics separately to overcome noise and reverberation. As a result, the interaction between the spectral normalization (e.g. mean and variance normalization, MVN) and temporal normalization (e.g. temporal structure normalization, TSN) is ignored. We propose a joint spectral and temporal normalization (JSTN) framework to simultaneously normalize these two aspects of feature statistics. In JSTN, feature trajectories are filtered by linear filters and the filters' coefficients are optimized by maximizing a likelihood-based objective function. Experimental results on Aurora-5 benchmark task show that JSTN consistently outperforms the cascade of MVN and TSN on test data corrupted by both additive noise and reverberation, which validates our proposal. Specifically, JSTN reduces average word error rate by 8-9% relatively over the cascade of MVN and TSN for both artificial and real noisy data.

Index Terms— robust speech recognition, feature normalization, temporal structure normalization, dereverberation.

1. INTRODUCTION

The performance of speech recognition degrades significantly when the test environment is different from the training environment [1]. The mismatch of environments is due to several factors, e.g. transmission channel, additive background noise, and reverberation. To reduce the mismatch, there are usually two approaches, i.e. the feature compensation/normalization approach [2, 3] and model adaptation approach [4, 5]. In this paper, we are interested in the feature normalization approach due to its advantages, e.g. no requirement for noise estimation, simple implementation, low computation overhead, and good performance on various tasks.

Feature normalization techniques normalize feature statistics to minimize the difference between the statistics of clean and noisy features. They can be classified into two groups, i.e spectral normalization and temporal normalization. In spectral normalization, the probability distribution of features are normalized to a reference distribution. As speech features such as Mel-frequency cepstral coefficients (MFCC) mainly capture the spectral information of speech in a short window (e.g. 25ms), we call this group of normalization spectral normalization. Spectral normalization methods differ in which aspect of feature distribution their normalize. The cepstral mean normalization (CMN) [6] and cepstral variance normalization (CVN) [7] normalize the mean and variance of the features, respectively. In addition, histogram equalization (HEQ) [3] normalizes the histogram of the features. On the other hand, in temporal normalization, the temporal structure of features are normalized. For example, temporal structure normalization (TSN) filters cepstral feature trajectories to normalize the modulation spectrum of speech [8, 9]. Similar normalization is also applied on filterbank trajectories to reduce reverberation effects [10]. As spectral and temporal normalization are complementary to each other, they can be applied together (usually in tandem) to achieve more robust features, e.g. MVN (mean and variance normalization, cascade of CMN and CVN) can be used with TSN to further improve robustness [9].

Although feature normalization methods are simple and effective, they suffer from two major limitations. First, the spectral and temporal normalization are combined in an ad-hoc manner by applying them in cascade. As the features distribution and temporal structure of features are related, simple combination of spectral and temporal normalization ignores the interactions between the two and results in suboptimal normalization. Second, feature normalization methods usually use a very simple reference model to represent the clean features. For example, in MVN the reference includes only 1 mean and 1 variance for each feature dimension. In TSN, the reference is just a modulation spectrum template for each dimension. Past research [11] shows that it is beneficial to use a more detailed reference model for feature normalization.

To address the limitations of feature normalization, we propose a framework for joint spectral and temporal normalization (JSTN) of features. In the JSTN framework, feature trajectories are filtered by finite impulse response (FIR) filters. The weights of the filter are optimized such that the filtered features will fit to both spectral and temporal reference models, which are trained from clean features. In addition, the spectral and temporal reference models are Gaussian mixtures models (GMM) and Gaussian models respectively, and can represent clean feature's statistics in more details.

The rest of the paper is organized as follows. In section 2, the details of JSTN framework is described. In section 3, the experimental results on Aurora-5 benchmark task is presented and discussed. In section 4, conclusions and future directions are presented.

2. JOINT SPECTRAL AND TEMPORAL NORMALIZATION

2.1. Overview

To simultaneously normalize both the temporal and spectral characteristics of features towards reference characteristics represented by

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

spectral and temporal reference models, several problems need to be solved, i.e. 1) What types of transform shall we use to process the features? 2) How to represent the spectral and temporal characteristics of the features? 3) How to define an objective function to include both the spectral and temporal normalization? and 4) How to obtain a solution for such an objective function? In the following sections, we will give an answer to these questions.

2.2. Linear Transform of Feature Trajectories

Linear transform is used in JSTN to process features. Unlike conventional feature space transforms (e.g. constrained maximum likelihood linear regression, CMLLR[12]) which transform feature vectors of individual frames, we transform the feature trajectories of each feature dimension cross multiple frames. This is equivalent to filtering feature trajectories with a FIR filter. The linear filtering of features is represented as

$$y_t^d = \sum_{\tau=0}^{2M} w_\tau^d x_{t-M+\tau}^d = \mathbf{w}_d^T \mathbf{x}_{dt}, \;\; d = 1, ..., D; \;\; t = 1, ..., T. \quad (1)$$

where x_t^d and y_t^d are the original and filtered features at frame t and dimension d, respectively. D is the dimension of feature vectors and T is the number of frames in the test utterance. $\mathbf{w}_d = [w_0^d, ..., w_{2M}^d]^T$ is the weight vector of the FIR filter for dimension d and the filter length is $2M + 1$. $\mathbf{x}_{dt} = [x_{t-M}^d, ..., x_{t+M}^d]^T$ is the input of the filter at frame t, dimension d. The reason for using linear transform of feature trajectories is that transforming feature trajectories is able to modify both the feature's temporal structure and distribution, while using linear transform of feature vectors can only modify the feature's distribution. The linear transformation of feature trajectories can be seen as a generalization of MVN. In MVN, the output y_t^d is only a scaled and shifted version of the input x_t^d.

2.3. Representation of Spectral and Temporal Information

The spectral information of speech is simply carried by the feature vectors. GMM is used as the spectral reference model to model the distribution of the feature vectors and trained from clean feature vectors.

To represent the temporal information of speech, one option is the speech modulation spectrum [13], i.e. the power spectrum density (PSD) function of the feature trajectories. In JSTN, we use the square root of PSD function to represent temporal information:

$$\mathbf{g}_x^d = |\text{DFT}(\mathbf{x}^d)| \quad (2)$$

where $\mathbf{g}_x^d = [g_x^d(1), ..., g_x^d(k), ..., g_x^d(K)]^T$, and $g_x^d(k)$ represents the square root PSD function at modulation frequency bin k for feature dimension d. K is the number of modulation frequency bins. DFT$(\cdot)$ represents discrete fourier transform (DFT). $|c|^2 = cc^*$ where c^* is the conjugate of c. The filtered square root PSD is

$$\mathbf{g}_y^d = \mathbf{g}_x^d \circ \mathbf{h}^d \quad (3)$$

where $\mathbf{h}^d = [h^d(1), ..., h^d(K)]^T$ is a $K \times 1$ vector with nonnegative elements that represents the magnitude response of the filter $\mathbf{w}_d$ and $\circ$ represents Hadamard product (i.e. entrywise product).

If the elements of $\mathbf{w}_d$ is arranged in certain way, there is a simple representation for $\mathbf{h}^d$. For example, if $\mathbf{w}_d$ is a Type I FIR filter [14], i.e. the filter is of odd order and its weights are symmetric w.r.t. the cental weight, the magnitude response of the filter is:

$$h^d(k) = \sum_{\tau=0}^{M} a_\tau^d \cos(\pi\tau k/K) = \mathbf{a}_d^T \mathbf{p}_k, \; k = 0, ..., K. \quad (4)$$

where $\mathbf{a}_d = [a_0^d, ..., a_M^d]^T$ is a vector of filter coefficients and $\mathbf{p}_k = [\cos(\pi 0k/K), \cos(\pi 1k/K), ..., \cos(\pi Mk/K)]^T$ is a constant vector. For Type I FIR filter, a_τ^d is related to w_τ^d as follows:

$$\begin{aligned} a_0^d &= w_M^d, \; \tau = 0 \\ a_\tau^d &= 2w_{M-\tau}^d = 2w_{M+\tau}^d, \; \tau > 0 \end{aligned} \quad (5)$$

The magnitude response can be represented as a vector $\mathbf{h}^d = \mathbf{P}^T \mathbf{a}_d$, where $\mathbf{P} = [\mathbf{p}_0, ..., \mathbf{p}_K]$ is a $(M + 1) \times K$ constant matrix. The filtered square root PSD in (3) is then rewritten as

$$\mathbf{g}_y^d = \mathbf{g}_x^d \circ \mathbf{P}^T \mathbf{a}_d \quad (6)$$

In (6), $\mathbf{g}_x^d$ and $\mathbf{P}$ are constants. The filtered square root PSD is a linear function of the filter's parameters $\mathbf{a}_d$.

2.4. Objective function

To find the weight vector $\mathbf{w}_d$ (or equivalently $\mathbf{a}_d$) that simultaneously normalize both the temporal and spectral characteristics of test features, we propose to maximize the following objective function:

$$\mathcal{O}(\mathbf{A}) = \frac{1}{T}\text{log}p(\mathbf{Y}|\lambda_s) + \frac{\alpha}{D}\text{log}p(\mathbf{G}|\lambda_g) \quad (7)$$

where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_D]$ is the matrix of filter weights, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_T]$ is the matrix of normalized features, $\mathbf{y}_t = [y_t^1, ..., y_t^D]^T$ is the filtered feature vector at frame t. $\mathbf{G} = [\mathbf{g}_y^1, ..., \mathbf{g}_y^D]$ is the matrix of filtered square root PSD functions. λ_s is the spectral reference model that represents the distribution of clean feature vectors and is a GMM in this study. λ_g is the temporal reference model that represents the distribution of square root PSD functions of clean feature trajectories. In this study, λ_g contains D GMMs, one for each feature dimension. λ_g can be trained from clean feature trajectories. With the objective function in (7), we optimize the filter weights by simultaneously maximizing the likelihood of filtered feature vectors on the spectral reference model and the likelihood of filtered square root PSD on the temporal reference model. α is used to balance the spectral and temporal normalization.

There are two advantages to perform spectral and temporal normalization jointly. First, if the temporal normalization is performed alone, only temporal information (represented by PSD functions) are normalized and the temporal structure of test features are made to be similar to that of clean features. This is similar to the TSN filter [9] which normalizes test features' PSD functions to clean PSD templates. However, there is no guarantee that the filtered features will fit well with the acoustic model. By introducing spectral normalization in JSTN, the filtered features will also have good fit with the spectral reference model, which represents the same clean features space as the acoustic model. Second, if the spectral normalization is performed alone, the maximum likelihood objective function will be ill-posed as the changes in the variance of the features (i.e. the Jacobian term) cannot be evaluated due to the fact that we are using linear filtering along temporal axis rather than linear transform of feature vectors (which is the case of CMLLR [12]). Such problem is also encountered in other ML-based feature space transformations where the Jacobian cannot be evaluated, e.g. in ML adaptation of HEQ (ML-HEQ) [11], voice conversion [15], and ML-based filtering for dereverberation [16]. In ML-HEQ, this problem is reduced by regularizing the parameters of the system, while in voice conversion, simultaneous maximizing the likelihood of the filtered features' variances is used. In ML filtering [16], the filtered features are simply variance normalized. The proposed JSTN goes a step further than [15], as we simultaneously maximize the likelihood of the PSD functions rather than just the variances.

2.5. Solution

The first term in the objective function is expanded as follows:

$$\frac{1}{T}\log p(\mathbf{Y}|\lambda_s) \quad = \quad \frac{1}{T}\sum_{t=1}^{T}\log\sum_{m=1}^{M} c_m\mathcal{N}(\mathbf{y}_t;\mu_m,\Sigma_m) \quad (8)$$

where c_m, μ_m, and Σ_m are the weight, mean, and diagonal covariance of the m^{th} Gaussian in λ_s, respectively. M denotes the number of Gaussians in λ_s. The second term of (7) is expanded as

$$\frac{\alpha}{D}\log p(\mathbf{G}|\lambda_g) \quad = \quad \frac{\alpha}{D}\sum_{d=1}^{D}\log\sum_{i=1}^{I}\tilde{c}_{id}\mathcal{N}(\mathbf{g}_y^d;\tilde{\mu}_{id},\tilde{\Sigma}_{id}) \quad (9)$$

where $\tilde{c}_{id}$, $\tilde{\mu}_{id}$, and $\tilde{\Sigma}_{id}$ are the weight, mean, and diagonal covariance of mixture i of the GMM for dimension d, respectively. I is the number of Gaussians per dimension.

As there is no closed-form solution for maximizing the objective function, expectation maximization (EM) algorithm is used. The auxiliary function of the first term of the objective function is

$$\mathcal{Q}_1(\mathbf{A},\hat{\mathbf{A}}) \quad = \quad -\frac{1}{2T}\sum_{tm}\gamma_m(t)\sum_{d=1}^{D}\frac{(\mathbf{w}_d^T\mathbf{x}_{dt}-\mu_{md})^2}{\sigma_{md}^2} \quad (10)$$

where $\hat{\mathbf{A}}$ is the filter weights estimated from previous iteration. $\gamma_m(t)$ is the posterior probability of mixture m. Terms not related to $\mathbf{A}$ are ignored in (10). By using the relationship between $\mathbf{w}_d$ and $\mathbf{a}_d$ in (5), we have the following equation:

$$\mathbf{w}_d^T\mathbf{x}_{dt} \quad = \quad a_0^d x_t^d + \sum_{\tau=1}^{M} a_\tau^d(x_{t+\tau}^d + x_{t-\tau}^d) = \mathbf{a}_d^T\mathbf{q}_{dt} \quad (11)$$

where $\mathbf{q}_{dt} = [x_t^d, (x_{t+1}^d + x_{t-1}^d), ..., (x_{t+\tau}^d + x_{t-\tau}^d)]^T$ is a $(M+1)\times 1$ vector. Then equation (10) can be rewritten as

$$\mathcal{Q}_1(\mathbf{A},\hat{\mathbf{A}}) \quad = \quad -\frac{1}{2T}\sum_{tm}\gamma_m(t)\sum_{d=1}^{D}\frac{(\mathbf{a}_d^T\mathbf{q}_{dt}-\mu_{md})^2}{\sigma_{md}^2} \quad (12)$$

For the second term of the objective function, we can use the following auxiliary function:

$$\begin{aligned}\mathcal{Q}_2(\mathbf{A},\hat{\mathbf{A}}) \quad = \quad & -\frac{\alpha}{2D}\sum_{di}\gamma_i(d)\left[\mathbf{g}_x^d\circ\mathbf{P}^T\mathbf{a}_d-\tilde{\mu}_{id}\right]^T \\ & \tilde{\Sigma}_{id}^{-1}\left[\mathbf{g}_x^d\circ\mathbf{P}^T\mathbf{a}_d-\tilde{\mu}_{id}\right] \quad (13)\end{aligned}$$

where $\gamma_i(d)$ is the posterior of mixture i in dimension d.

The solution of $\mathbf{A}$ can be obtained by maximizing the total auxiliary function $\mathcal{Q}(\mathbf{A},\hat{\mathbf{A}}) = \mathcal{Q}_1(\mathbf{A},\hat{\mathbf{A}}) + \mathcal{Q}_2(\mathbf{A},\hat{\mathbf{A}})$ w.r.t. $\mathbf{a}_d$ for each feature dimension independently. Take the partial differentiation of the auxiliary function w.r.t. $\mathbf{a}_d$, after some manipulations, we get

$$\frac{\partial\mathcal{Q}(\mathbf{A},\hat{\mathbf{A}})}{\partial\mathbf{a}_d} \quad = \quad -\mathbf{B}_d\mathbf{a}_d + \mathbf{c}_d - \alpha\mathbf{D}_d\mathbf{a}_d + \alpha\mathbf{e}_d = 0 \quad (14)$$

where

$$\mathbf{B}_d \quad = \quad \frac{1}{T}\sum_{tm}\gamma_m(t)\frac{\mathbf{q}_{dt}\mathbf{q}_{dt}^T}{\sigma_{md}^2} \quad (15)$$

$$\mathbf{c}_d \quad = \quad \frac{1}{T}\sum_{tm}\gamma_m(t)\frac{\mu_{md}\mathbf{q}_{dt}}{\sigma_{md}^2} \quad (16)$$

$$\mathbf{D}_d \quad = \quad \frac{1}{D}\sum_{i}\gamma_i(d)\sum_{k=1}^{K}\frac{(g_{xk}^d)^2\mathbf{p}_k\mathbf{p}_k^T}{\tilde{\sigma}_{idk}^2} \quad (17)$$

$$\mathbf{e}_d \quad = \quad \frac{1}{D}\sum_{i}\gamma_i(d)\sum_{k=1}^{K}\frac{g_{xk}^d\tilde{\mu}_{idk}\mathbf{p}_k}{\tilde{\sigma}_{idk}^2} \quad (18)$$

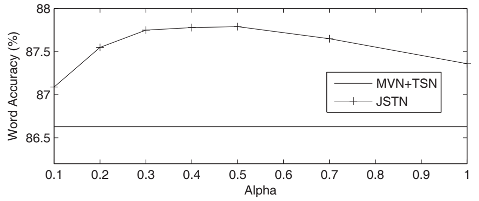

Fig. 1. Performance of JSTN on meeting data with different α.

are the accumulated statistics for dimension d. The closed-form solution for $\mathbf{a}_d$ is

$$\hat{\mathbf{a}}_d \quad = \quad (\mathbf{B}_d + \alpha\mathbf{D}_d)^{-1}(\mathbf{c}_d + \alpha\mathbf{e}_d) \quad (19)$$

The filter weights $\mathbf{a}_d$ is optimized for each feature dimension iteratively using (19).

3. EXPERIMENTS

3.1. Experimental Settings

We evaluate the proposed JSTN method on Aurora-5 task [17], which is used to evaluate the robustness of speech recognition against channel, additive noise, and reverberations. Aurora-5 is a English continuous digit string recognition task. Two test sets used in this study are the "living room" and "office" sets, which are simulated reverberated speech. The reverberation times for "living room" and "office" are randomly selected from 0.4 to 0.5s and from 0.3 to 0.4s, respectively. The noisy version of the reverberated speech are tested, including four signal to noise ratios (SNR), i.e. 15dB, 10dB, 5dB, and 0dB. The corrupting noises are real world noises, including shopping mall, restaurant, exhibition hall, office, and hotel lobby noises. Besides artificially generated test data, we also test on real meeting recordings in Aurora-5. Speech signals were simultaneously recorded by 4 hands-free microphones and corrupted by both reverberation and a small amount of background noise. The 4 microphones were placed in different locations of the meeting room and capture different reverberation effects.

In our experiments, the training of acoustic models follows the standard configuration in [17] and clean condition training is used. MFCC features augmented by their first and second derivatives are used. C0 is used instead of log energy. The same feature normalization method is always applied to training and testing features, except for JSTN. In JSTN, the training features are processed by MVN+TSN (MVN followed by TSN) and the test features are processed by JSTN. This is because we haven't studied adaptive training using JSTN yet. The spectral reference model λ_s is built by pooling the 716 Gaussians in the acoustic model. In the temporal reference model, a single Gaussian (i.e. $I = 1$) is used to model the distribution of the square root PSD for each feature dimension, and is trained from MVN processed clean training features. The filter in JSTN is initialized by TSN filter, and both filters have 33 taps. Following [9], the PSD functions of feature trajectories are obtained by using the Yule-Walker method with order 6 and K is empirically set to 22.

3.2. Experimental Results

Fig. 1 shows the performance of JSTN with different α on the meetings data averaged over 4 microphones. It is observed that performance is stable when α is around 0.4. Therefore, in the following experiments, α is set to 0.4.

Table 1. Recognition accuracy on living room and office data. Raw represents the results of raw MFCC featuers without any feature normalization. Clean refers to matched training-testing case. Office and Living room denote reverberated speech without additive noise. Avg refers to average over all the test cases. RR stands for relative error rate reduction achieved by JSTN over MVN+TSN.

SNR	Raw	MVN	MVN+TSN	JSTN	RR(%)
Clean	99.34	99.38	99.38	99.34	-6.5
Office	93.55	93.86	94.26	96.18	33.4
Office 15dB	70.61	81.00	83.99	88.23	26.5
Office 10dB	43.23	68.79	75.61	79.78	17.1
Office 5dB	18.51	49.89	61.65	64.60	7.7
Office 0dB	7.56	28.17	41.69	42.06	0.6
Living room	82.49	81.30	83.07	88.06	29.5
Living room 15dB	53.91	62.62	68.62	75.59	22.2
Living room 10dB	29.99	50.87	60.37	65.64	13.3
Living room 5dB	12.94	35.76	48.77	50.78	3.9
Living room 0dB	6.65	21.11	33.20	32.54	-1.0
Avg	41.94	57.34	65.12	68.35	9.2

Table 2. Recognition accuracy for real meeting data. 6, 7, E, and F are the 4 different microphones used to recorded the data.

Mic.	Raw	MVN	MVN+TSN	JSTN	RR(%)
6	78.56	87.12	89.18	90.26	10.0
7	63.60	82.28	85.97	86.65	4.8
E	73.42	80.65	84.14	85.15	6.4
F	80.29	85.67	87.21	88.93	13.4
Avg	73.97	83.93	86.63	87.75	8.4

The performance of JSTN and other feature normalization methods on the artificial data is shown in Table 1. From the table, we can see that the MVN (spectral normalization) produces significant improvement over the baseline result for most of the test cases. Furthermore, MVN+TSN, i.e. the cascade of spectral and temporal normalization, produces significantly better results than MVN alone. This shows that it is beneficial to apply both spectral and temporal normalization. In addition, the JSTN outperforms MVN+TSN significantly, especially in high SNR levels (except clean case). This shows that the joint normalization of spectral and temporal information of features is better than the cascade of the two normalizations. The relatively small improvement in low SNR levels is probably due to that the posterior probabilities $\gamma_m(t)$ are not accurate at low SNR.

We also test JSTN on real meeting recordings and the results are shown in Table 2. The data here are recorded in reverberation environment with small amount of background noise. The results show that JSTN consistently outperform MVN+TSN and yields a 8.4% relative average word error rate reduction. This shows that JSTN also works well for reverberated speech recorded in real world.

4. CONCLUSIONS

In this paper, we proposed a framework for joint spectral and temporal normalization of feature statistics for robust speech recognition. Experimental study on Aurora-5 tasks show that the JSTN produces better results than simply applying spectral normalization (MVN) and temporal normalization (TSN) in cascade for recognizing both additive noise and reverberation corrupted speech. The JSTN framework is a flexible way for feature normalization and can be extended in various ways. For example, currently the temporal reference model only captures the global temporal information of test utterances. More detailed temporal information (e.g. local temporal information) is worthy a study in the future.

5. REFERENCES

[1] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, no. 3, pp. 261–291, 1995.

[2] P. J. Moreno, B. Raj, and R. M. Stern, "Data-driven environmental compensation for speech recognition: A unified appraoch," *Speech Communication*, vol. 24, no. 4, pp. 267–285, Jul. 1998.

[3] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005.

[4] 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.

[5] H.-G. Hirsch and H. Finster, "A new approach for the adaptation of hmms to reverberation and background noise," *Speech Communication*, vol. 50, pp. 244–263, March 2008.

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

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

[8] X. Xiao, E. S. Chng, and H. Li, "Temporal structure normalization of speech feature for robust speech recognition," *IEEE Signal Processing letters*, vol. 14, no. 7, pp. 500–503, 2007.

[9] ——, "Normalization of the speech modulation spectra for robust speech recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1662–1674, Nov. 2008.

[10] X. Lu, M. Unoki, and S. Nakamura, "Sub-band temporal modulation envelopes and their normalization for automatic speech recognition in reverberant environments," *Computer Speech and Language*, vol. 25, no. 3, pp. 571 – 584, 2011.

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

[12] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[13] L. Atlas and S. A. Shamma, "Joint acoustic and modulation frequency," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 7, pp. 668–675, 2003.

[14] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice-Hall, 1999.

[15] 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.

[16] K. Kumar and R. Stern, "Maximum-likelihood-based cepstral inverse filtering for blind speech dereverberation," in *Proc. ICASSP '10*, Dallas, Texas, USA, Apr. 2010, pp. 4282 –4285.

[17] H. G. Hirsch, "Aurora-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments," Tech. Rep., 2007.