NORMALIZING THE SPEECH MODULATION SPECTRUM FOR ROBUST SPEECH RECOGNITION

Xiong Xiao†‡, Eng Siong Chng†, Haizhou Li†‡

†School of Computer Engineering, Nanyang Technological University, Singapore
‡Institute for Infocomm Research, Singapore

ABSTRACT

This paper presents a novel feature normalization technique for robust speech recognition. The proposed technique normalizes the temporal structure of the feature to reduce the feature variation due to environmental interferences. Specifically, it normalizes the utterance-dependent feature modulation spectrum to a reference function by filtering the feature using a square-root Wiener filter in the temporal domain. We show experimentally that the proposed technique when combined with mean and variance normalization technique (MVN) reduces the word error rate significantly on the AURORA-2 task, with relative error rate reduction 69.11% compared to the baseline.

Index Terms— Speech recognition, feature normalization, modulation spectrum, square-root Wiener filter, temporal filter

1. INTRODUCTION

Current automatic speech recognition (ASR) systems are not robust against environmental interferences, such as additive background noise and convolutional channel distortions. The robustness problem is attacked by various techniques and one group of them are the normalization techniques that aim to normalize the statistics of the speech feature.

There are three techniques for normalizing the statistical distribution of the feature, they are the cepstral mean normalization (CMN) technique [1], the cepstral variance normalization (CVN) technique [2], and the histogram equalization (HEQ) technique [3]. The CMN technique removes the utterance mean of the feature to reduce channel distortions. The CVN technique normalizes the feature variance to a fixed value to put the features in the same scale. The CMN and CVN are usually used in cascade to form the mean and variance normalization (MVN) to normalize both the first-order and the second-order moments of the feature. The HEQ technique equalizes the histogram of the speech feature to a fixed probability distribution function (pdf), such as Gaussian distribution.

The above-mentioned techniques focus on normalizing the statistical distribution of the speech features. Recent researches [4]-[7] show that it is also desirable to filter the feature in the temporal domain to improve the robustness of ASR. An early temporal filter called the representations relative spectra (RASTA) [4] is a band-pass infinite impulse response (IIR) filter that operates in the log filterbank domain. The passband of the RASTA filter is from 0.26 to 14.3Hz in modulation frequency. The attenuation of the low frequency reduces the convolutional noise's effect in the similar way as the CMN, and the attenuation of the high frequency reduces feature variations due to the feature extraction process. The RASTA filter was reported to perform well on reducing the effect of convolutional noises, but it is less effective in removing the effect of additive noises. Later, the RASTA filter is supported by the experimental observation of Kanedera et al.[5] which shows the relative importance of different bands of modulation frequency on ASR. Their observation shows that the low frequency 0-1Hz and high frequency 16-50Hz are harmful or not useful for ASR, while the modulation frequency around 4Hz is most useful for recognition task.

Besides the empirically designed RASTA filter, several data-driven temporal filters have been designed from the speech data using some criteria, such as linear discriminant analysis (LDA), principle component analysis (PCA) and minimum classification error (MCE), and they are summarized nicely in [6]. The basic idea of these filters is to project the speech feature into a subspace for enhanced discriminative ability, and the filters are mostly low-pass or band-pass. In [6], these filters are reported to improve the recognition accuracy significantly for a connected Chinese digital string task.

Another technique called MVA [7] is the cascade of the MVN and a low-pass autoregressive moving average (ARMA) filter. The MVA technique is motivated by the observation that the noisy speech features after MVN operation are usually less smooth than their clean counterpart. Despite its simplicity, the MVA achieves significant improvement on recognition accuracy for the AURORA-2 task.

The temporal filters described above focus on smoothing the speech feature rather than explicitly normalizing the temporal structure of the feature. In this paper, we will examine the normalization strategy and investigate its effect on the ASR. Specifically, we design temporal filters to normalize the utterance-dependent feature PSD (i.e. the modulation spectrum) to a reference PSD function, which in effect is to normalize the feature's temporal structure. The square-root Wiener filter is used for the normalization and two feature post-processing schemes are proposed based on it.

This paper is organized as follows. In section 2, we introduce the proposed temporal structure normalization (TSN) technique. In section 3, we present the experimental results and compare the TSN with existing filters. Finally, we conclude in section 4.

2. NORMALIZING THE PSD OF SPEECH FEATURE

2.1. Temporal Structure Varies with Environment

We use the Mel-scaled filterbank cepstral coefficient (MFCC) as the feature for speech recognition. Let $x(n,k)$ be the cepstral coefficient of the n^{th} frame and k^{th} MFCC channel (the k^{th} feature) of an utterance. Let the coefficients of all frames for the k^{th} channel be $x_k(n)$. Hence, there are K time series, $x_1(n)$ to $x_K(n)$, where K is the number of channels. For our experiments, the raw features are the $c0 - c12$ cepstral coefficients, delta and acceleration coefficients, thus $K = 39$. The time series of these raw features are first processed by MVN, then normalized by the proposed temporal structure normalization (see Fig. 1). After MVN, these time series

1-4244-0728-1/07/$20.00 ©2007 IEEE

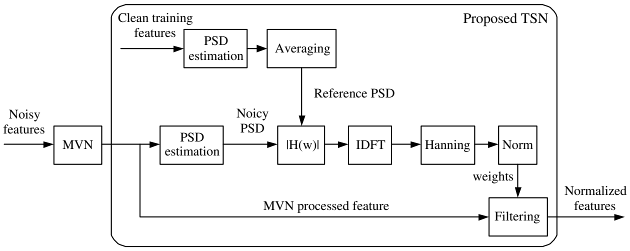

Fig. 1. The block diagram of the proposed framework

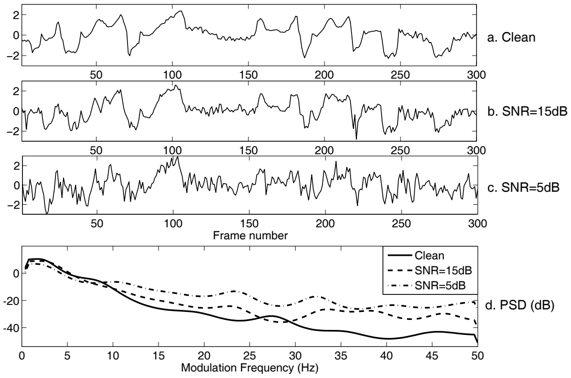

Fig. 2. An example of the time series of MFCC feature $c1$ corrupted by additive car noise after MVN operation in three SNR levels.

will have zero mean and unit variance. The sampling rate of the time series, also known as feature frame rate, is 100 Hz. We will examine how the temporal structure of these time series in the form of PSD changes with noise.

Fig. 2 shows the time series for $c1$ of one utterance corrupted by additive car noise after MVN and its PSD in three SNR levels. The PSD is estimated using the Yule-Walker method with the order of the model being 15. From Fig. 2(a)-(c), it is obvious that the speech features in low SNR level are less smooth than those in high SNR levels. In the PSD diagram Fig. 2(d), the noisy features' PSD functions have a higher power density in the high modulation frequency ranges than their clean counterpart. This explains our previous observation. To reduce the feature's variation due to noise, we propose to normalize the feature's PSD function.

2.2. Normalizing the Temporal Structure Using the Square-root Wiener Filters

It is known that the square-root Wiener filter has the property of normalizing the filtered signal's PSD to the desired signal's PSD [8]. In the following text, we will introduce our approach of applying the square-root Wiener filter to normalize the speech feature's PSD functions.

Let $y_k(n)$ be the observed noisy speech feature series for the k^{th} channel and $x_k(n)$ be its underlying clean version. Let $v_k(n)$ be the noise that is the difference between $x_k(n)$ and $y_k(n)$. Hence, we have the following relationship:

$$y_k(n) = x_k(n) + v_k(n), \quad \text{for } k = 1, ..., K \tag{1}$$

Let $P^k_{xy}(\omega)$ be the cross PSD between $x_k(n)$ and $y_k(n)$, and let $P^k_{xx}(\omega)$, $P^k_{yy}(\omega)$ and $P^k_{vv}(\omega)$ be the PSD of $x_k(n)$, $y_k(n)$ and $v_k(n)$, respectively. The magnitude response of the square-root Wiener filter is [8]

$$|H_k(\omega)| = \sqrt{P^k_{xy}(\omega)/P^k_{yy}(\omega)} \tag{2}$$

If $x_k(n)$ and $v_k(n)$ are assumed to be statistically independent from each other, $P^k_{yy}(\omega) = P^k_{xx}(\omega) + P^k_{vv}(\omega)$ and $P^k_{xy}(\omega) = P^k_{xx}(\omega)$. Although this assumption is not completely true, features that have been post-processed by the square-root Wiener filter showed improvement in recognition accuracy. The square-root Wiener filter can be rewritten as

$$|H_k(\omega)| = \sqrt{P^k_{xx}(\omega)/P^k_{yy}(\omega)} \tag{3}$$

From equation (3), we find that the square-root Wiener filter depends on the PSD of the noisy feature and clean feature. For implementation, the noisy PSD $P^k_{yy}(\omega)$ can be estimated from a short segment of $y_k(n)$, such as an utterance, by assuming that $y_k(n)$ is stationary in the segment. As the PSD of the clean feature $P^k_{xx}(\omega)$ is unknown, we instead use an averaged PSD function $\bar{P}^k_{xx}(\omega)$ to evaluate $|H_k(\omega)|$. We call $\bar{P}^k_{xx}(\omega)$ the reference PSD functions and obtain them by averaging the clean feature's PSD over multiple utterances. This is possible as the PSD of clean features of a channel is similar for different utterances. To reduce the variation of $\bar{P}^k_{xx}(\omega)$ due to the effects of speech content and speaker, we average the feature PSD over a collection of different utterances.

It is easy to integrate our technique with other temporal filtering techniques. For example, we can combine the MVA technique and our method by either filtering the features using MVA before training of the reference PSD functions $\bar{P}^k_{xx}(\omega)$, or multiplying the magnitude response of ARMA filter to $|H_k(\omega)|$. In either way, the resulting filters will not only normalize the feature's PSD, but also low-pass filter the feature.

After obtaining $|H_k(\omega)|$, the filter's coefficients can be found using the windowed FIR filter design method [9]. We summarize the proposed method as follows (see also Fig.1).

Training the reference PSD functions:

1. Calculate the feature PSD of all channels of all training utterances $P^{k,m}_{xx}(\omega)$, for $k = 1, ..., K$ and $m = 1, ..., M$, where M is the number of utterances used for training $\bar{P}^k_{xx}(\omega)$
2. Find the averaged clean PSD function using

$$\bar{P}^k_{xx}(\omega) = \frac{1}{M}\sum_{m=1}^{M} P^{k,m}_{xx}(\omega), \quad k = 1, ..., K \tag{4}$$

Designing the FIR filters:

1. For each incoming utterance, calculate the utterance's feature PSD $P^k_{yy}(\omega)$ for $k = 1, ..., K$.
2. Find the $|H_k(\omega)|$ for $k = 1, ..., K$ using

$$|H_k(\omega)| = \sqrt{\bar{P}^k_{xx}(\omega)/P^k_{yy}(\omega)}, \quad \omega \in [-\pi, \pi] \tag{5}$$

3. Find the filter's weights using the inverse discrete Fourier transform (IDFT).

$$w_k(i) = \text{IDFT}(|H_k(\omega)|) \tag{6}$$

4. Form the filter's weights $w'_k(i)$ using only the middle part of $w_k(i)$ to reduce the filter length and computational complexity. This is possible because the most significant weights are concentrated in the middle of $w_k(i)$.

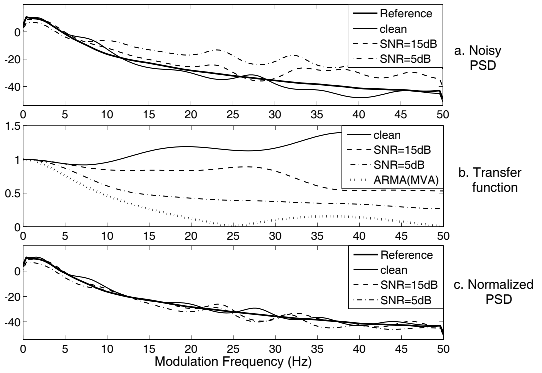

Fig. 3. Effect of TSN1 on the feature's PSD as appeared in Fig. 2(d)

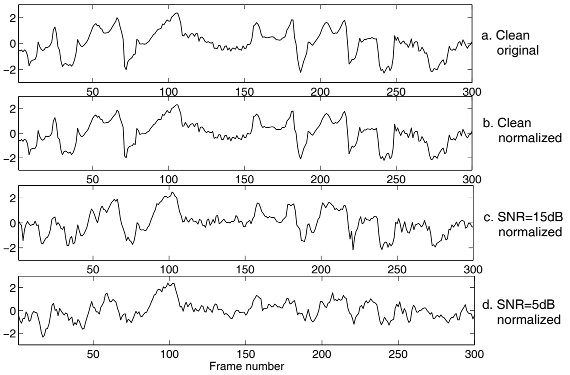

Fig. 4. Effect of TSN1 on the feature as appeared in Fig. 2(a)-(c)

5. Apply Hanning window on $w'_k(i)$ to reduce truncation effect.
6. Normalize the sum of the weights to one to ensure that the filter's gain is unity at zero frequency.

The phase responses of the filters designed using the above windowed method are linear and identical for all channels and utterances. After the filters are designed, the normalized features are estimated as $\hat{x}_k(n) = y_k(n) \otimes w'_k(i)$ for $k = 1, ..., K$, where $\otimes$ denotes the convolution operation.

3. EXPERIMENTS AND RESULTS

3.1. Experiment Settings

For our experiments, the PSD is estimated using the Yule-Walker method with the autoregressive model order be 15 to obtain the desired smoothness for the PSD. The number of bins for the two sided PSD is 256 to ensure sufficient samples in the frequency domain. We experimented with two temporal structure normalization schemes, the TSN1 and TSN2. They differ in the features used to train the reference PSD functions: a) the training features for TSN1's reference functions are processed by MVN only, and b) those for TSN2 are first processed by MVN and then smoothed by the ARMA filter used in MVA [7]. The order M of the ARMA filter is experimentally decided to be 3 for our settings which results in the highest recognition accuracy. We use 1000 utterances from the training set of AURORA-2 database [10] for the training of the reference functions, with half male speakers and half female speakers. The filter length of 21 is chosen as it achieves the highest speech recognition accuracy for the AURORA-2 task.

3.2. Normalization Effect on Features

In Fig. 3-4, we show the normalization effects of TSN1. Fig. 3(a) shows the same PSD functions as that appeared in Fig. 2(d) along with the reference PSD function. Fig. 3(b) illustrates the magnitude response of the filters to equalize the noisy PSD to the reference PSD for different SNR levels. For the clean case, the magnitude response is slightly high-pass while for the SNR=15dB and 5dB cases, the filters are both low-pass. By comparing the magnitude responses of the TSN filters and the ARMA (order=3) filter, we find that the stop-band attenuation of TSN filters is much weaker than that of the ARMA filter. Fig. 3(c) shows that the normalized PSD functions are very similar to the reference PSD function.

Fig. 4(b)-(d) compares the normalized version of the time series of Fig. 2(a)-(c) with their original clean version in Fig. 4(a). It is observed that the

smoothness of the normalized features in different SNR levels is more similar and that the normalized clean features are almost the same as the original clean features. This indicates that the filtering process does not alter the clean features significantly. On the other hand, the normalized SNR=15dB and 5dB features are much smoother than their original features. This shows that the TSN1 filters the features of different SNR differently.

The normalization effects of TSN2 are similar to that of TSN1 except that the normalized features are more smoother in all SNR levels.

3.3. Recognition Results

In this section, we compare the performance of the proposed normalization schemes with 4 other methods using the AURORA-2 framework [10]. The training and testing of the recognition engine follow the scripts provided by the framework, except that the $c0$ is used, rather than the log energy. In all the experiments, the 13 MFCC features, $c0-c12$, together with their delta and acceleration features are generated prior to any post-processing. After these 39 features are generated, different post-processing techniques are applied on them separately. There are altogether six post-processing techniques, they are:

a) MVN: CMN followed by CVN.

b) RASTA: MVN followed by RASTA filtering.

c) MVA (M=3): MVN followed by ARMA filtering.

d) LPF: MVN followed by LPF filtering.

e) TSN1: MVN followed by TSN1.

f) TSN2: MVN followed by TSN2.

Among these techniques, the CMN and CVN are implemented utterancewise. The RASTA filter is implemented using the equation (1) in [4], with the pole value set to 0.94 for better performance. The LPF is a low-pass filter designed in the same way as the TSN filter except that its magnitude response is an ideal low-pass filter and identical for all channels. The filter length of LPF is set to be the same as the TSN filter and its optimal cut-off frequency is experimentally found to be 12Hz that yields the best recognition accuracies.

Table 1. Recognition Accuracy (%) for AURORA-2 Task Averaged Across the SNR Between 0 and 20 dB. RI (%) Is the Relative Error Rate Reduction Over the Baseline

Method	Set A	Set B	Set C	Avg.	RI
Baseline	53.17	47.89	63.05	53.03	-
MVN	77.91	79.48	77.70	78.49	54.20
RASTA	81.06	82.69	81.71	81.84	61.34
MVA	84.18	85.16	84.28	84.59	67.19
LPF	83.67	85.34	84.05	84.41	66.81
TSN1	84.27	85.87	83.62	**84.78**	67.60
TSN2	84.72	86.59	84.80	**85.49**	69.11

Table 2. Recognition Accuracy (%) for AURORA-2 Task for Each SNR Level Averaged Across Ten Noise Cases.

Method	Clean	20dB	15dB	10dB	5dB	0dB	-5dB
MVN	99.12	97.46	94.92	88.41	71.51	40.17	16.08
RASTA	99.10	97.27	94.94	89.60	76.50	50.89	22.30
MVA	99.10	97.81	95.95	91.38	80.43	57.39	27.09
LPF	99.23	97.92	96.07	91.46	79.94	56.69	26.27
TSN1	99.23	97.69	96.01	91.55	80.95	57.71	27.16
TSN2	99.26	97.93	96.13	92.06	81.76	59.56	28.13

The experimental results are summarized in Table 1-2. In Table 1, the MVN result shows that the normalization of the first and second order moments of the feature improves the accuracy significantly over the baseline and there is a relative error rate reduction of 54.20%. The RASTA, MVA and LPF all further improve the performance by filtering out some feature variations between the clean and noisy features, with the improvements of MVA and LPF noticeably higher than that of RASTA. The proposed TSN1 scheme produces slightly higher accuracy than the MVA and LPF. Finally, the TSN2 scheme achieves the highest accuracy among all the techniques and its improvement is higher than that of MVA by a 0.9% absolute improvement in accuracy.

In Table 2, we compare the performance of the post-processing techniques in difference SNR levels. It is observed that TSN2 outperforms all other results in all SNR levels.

3.4. Discussion

The proposed temporal structure normalization technique aims to normalize the feature's PSD function rather than simply smoothing the features. The TSN1 scheme provides only the normalization of the feature PSD, while the TSN2 scheme also provides extra smoothing. The extra smoothing enables the TSN2 scheme to outperform the TSN1 scheme in terms of recognition accuracy. This agrees with the experimental finding of Kanedera et al. [5] that stated that the high modulation frequency is not useful for speech recognition. Although the TSN1 scheme has poorer performance than the TSN2 scheme due to its mild smoothing, it preserves more speech details that may be useful for large vocabulary speech recognition tasks. We are going to examine the effect of smoothing on the speech recognition accuracy on large vocabulary tasks.

Our experimental results also showed that the proposed TSN schemes are superior to the fixed optimal low-pass filter with the same filter complexity, due to their ability to adapt the filter weights to different noise and SNR situations.

The extra computational cost introduced by our approach is on the IDFT and Yule-Walker PSD estimation operations which can both be implemented using efficient algorithms, such as the fast Fourier transform (FFT) and the Levinson-Durbin recursion [9], respectively. While there is an extra computational need, the performance improvement justifies the relatively modest computing overhead.

4. CONCLUSION

In this paper, we examined a new feature post-processing technique that normalizes the speech feature's temporal structure explicitly in the form of PSD. The proposed TSN technique filters the speech features aiming to bring the feature PSD functions of the current utterance to reference functions. Experimental results show that TSN improves the recognition accuracy for both clean and noisy cases for the AURORA-2 task.

5. REFERENCES

[1] S. Furui, "Cepstral analysis technique for automatic speaker verification", *IEEE Trans. Acoust., Speech, Signal Process*, vol. 29, no. 2, pp. 254-272, 1981

[2] O. Viikki, D. Bye, and K. Laurila, "A recursive feature vector normalization approach for robust speech recognition in noise", *In Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1998, vol. 11, pp. 733-736

[3] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Perez-Cordoba, M. C. Benitez and A. J. Rubio, "Histogram equalization of speech representation for robust Speech Recognition", *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 355-366, 2005

[4] H. Hermansky and N. Morgan, "RASTA processing of speech", *IEEE Trans. Speech, Audio Proceess.*, vol. 2, no. 4, pp. 578-589, 1994

[5] N. Kanedera, T. Arai, H. Hermansky, and M. Pavel "On the relative importance of various components of the modulation spectrum for automatic speech recognition", *Speech Commnunication*, vol. 28, No. 1, pp. 43-55, 1999

[6] J.-W. Hung and L.-S. Lee, "Optimization of temporal filters for constructing robust features in speech recognition", *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no 3, pp. 808-832, May 2006

[7] C.-P. Chen, J. Blimes and K. Kirchhoff, "Low-resource noise-robust feature post-processing on AURORA 2.0", In *Proc. ICSLP* 2002, pp. 2445-2448

[8] S. V. Vaseghi, "Advanced digital signal processing and noise reduction", 2nd Edition, John Wiley & Sons, LTD, 2000

[9] J. G. Proakis and D. G. Manolakis, "Digital signal processing-principles, algorithms, and applications", 3rd Edition, Prentice-Hall, 1996

[10] D. Pearce and H.-G. Hirsch, "The AURORA experimental framework for the performance evaluation of speech recogntion systems under noisy conditions", *In Proc. ICSLP* 2000, pp. 29-32