Computational Linguistics and Chinese Language Processing
Vol. 14, No. 1, March 2009, pp. 105-132

Study of Associative Cepstral Statistics Normalization Techniques for Robust Speech Recognition in Additive Noise Environments

Wen-Hsiang Tu* and Jeih-weih Hung*

Abstract

Feature statistics normalization techniques have been shown to be very successful in improving the noise robustness of a speech recognition system. In this paper, we propose an associative scheme in order to obtain a more accurate estimate of the statistical information in these techniques. By properly integrating codebook and utterance knowledge, the resulting associative cepstral mean subtraction (A-CMS), associative cepstral mean and variance normalization (A-CMVN), and associative histogram equalization (A-HEQ) behave significantly better than the conventional utterance-based and codebook-based versions in additive noise environments. For the Aurora-2 clean-condition training task, the new proposed associative histogram equalization (A-HEQ) provides an average recognition accuracy of 90.69%, which is better than utterance-based HEQ (87.67%) and codebook-based HEQ (86.00%).

Keywords: Speech Recognition, Noise-Robust Feature, Codebook

1. Introduction

The performance of a speech recognition system is often severely degraded when there is a mismatch between the acoustic conditions of the training and the application environments. This mismatch may come from various sources, such as additive noise, channel distortion, different speaker characteristics, and different speaking modes. A variety of robustness techniques with demonstrated improvement in system performance have been proposed to reduce this mismatch. For the purpose of handling additive noise, these robustness techniques can be roughly divided into three classes: adaptation of the speech models in the recognizer to make them better match the noise conditions, enhancement of the speech features before they are fed to the recognizer, and utilization of a noise robust representation of speech signals. In

* Dept of Electrical Engineering, National Chi Nan University, Nantou County, Taiwan, Republic of China
E-mail: aero3016@ms45.hinet.net; jwhung@ncnu.edu.tw

[Received January 13, 2009; Revised May 13, 2009; Accepted May 19, 2009]

the first class of approaches, compensation is performed on the pre-trained recognition model parameters so that the modified recognition models can more effectively classify the mismatched testing speech features collected in the application environment. Typical examples of this class include the well-known noise masking (Holmes & Sedgwick, 1986; Klatt, 1979; Nadas, Nahamoo, & Picheny, 1988), speech and noise decomposition (SND) (Varga & Moore, 1990), hypothesized Wiener filtering (Berstein & Shallom, 1991; Beattie & Young, 1992), vector Taylor series (VTS) (Acero, Deng, Kristjansson, & Zhang, 2000), maximum likelihood linear regression (MLLR) (Leggester & Woodland, 1995), model-based stochastic matching (Sankar & Lee, 1996; Lee, 1998), statistical re-estimation (STAR) (Moreno, Raj, & Stem, 1996), and parallel model combination (PMC) (Gales & Young, 1993; 1995a; 1995b). In the second class of approaches, the obtained testing speech features are modified in order to fit the acoustic conditions of pre-trained recognition models more compatibly. Examples of this class include the well-known spectral subtraction (SS) (Boll, 1979), codeword-dependent cepstral normalization (CDCN) (Acero, 1990), feature-based stochastic matching (Sankar & Lee, 1996; Lee, 1998), vector Taylor series (Segura, Benitez, de la Torre, Dupont, & Rubio, 2002; Moreno, Raj, & Stem, 1998), multivariate Gaussian-based cepstral normalization (RATZ) (Moreno, Raj, & Stem, 1996), and stereo-based piecewise linear compensation for environments (SPLICE) (Deng, Acero, Jiang, Droppo, & Huang, 2001; Droppo, Deng, & Acero, 2001). In the third class of approaches, a special robust speech feature representation is developed to reduce the sensitivity to various acoustic conditions; one way to develop this new feature representation is to normalize the statistics of the original speech features in both training and testing conditions in order to reduce the mismatch caused by noise. These feature statistics normalization techniques include cepstral mean subtraction (CMS) (Atal, 1974), cepstral mean and variance normalization (CMVN) (Tibrewala & Hermansky, 1997), cepstral gain normalization (CGN) (Yoshizawa, Hayasaka, Wada, & Miyanaga, 2004), histogram equalization (HEQ) (Hilger & Ney, 2006), higher-order cepstral moment normalization (HOCMN) (Hsu & Lee, 2004), cepstral shape normalization (CSN) (Du & Wang, 2008) *etc*. A common advantage of these methods is simplicity of implementation, since all of them focus on the front-end speech feature processing without the need of changing the back-end model training and recognition schemes. Regardless of the simplicity, these methods usually improve the recognition performance significantly under a noise-corrupted application environment.

A key process for most of the above normalization methods is to estimate the statistical information of speech features. For example, the first-order moment (mean), the first and second-order moments (mean and variance), and the probability distribution of features are required for CMS, CMVN, and HEQ, respectively. In most cases, the required statistical information is directly evaluated from the entire frame set of an utterance. Although simple in

implementation, the resulting utterance-based methods likely have some inherent drawbacks. First, they cannot be realized in an on-line manner since the computation and normalization of the statistics cannot be performed until the last frame of an utterance is received. Second, the number of frames in an utterance influences the accuracy of the obtained statistics. Third, since the length, or the number of different acoustic units, may vary from utterance to utterance, the normalized features of the same acoustic unit in an utterance may differ from those in another utterance.

In our previous works (Hung, 2006; 2008), we proposed that the statistics of features be evaluated based on two codebooks, named "pseudo stereo codebooks". Construction of the codebook of clean speech cepstra can occur off-line and prior to recognition. The codebook of noise-corrupted speech cepstra for each testing utterance is constructed by properly integrating the clean-speech codebook and the noise estimates, which often can be extracted from the first several frames of the utterance. The resulting codebook-based methods are expected to obtain more accurate estimate of feature statistics, and they can be implemented in an almost on-line manner. In (Hung, 2008), we have shown that codebook-based CMS and CMVN outperform conventional utterance-based ones in recognition accuracy for additive noise environments.

The original procedures in constructing the codebooks in (Hung, 2008), however, are somewhat simple, which possibly results in a less accurate estimate of the statistics for speech features. First of all, the clean speech codebook is built with all the feature vectors in the clean speech utterances for training. Since these utterances may contain quite long non-speech (silence) segments, it is likely that numerous codewords in the codebook just correspond to these non-speech parts. Second, the feature statistics are estimated by *uniformly* averaging the codewords, which ignores the relative significance of each codeword. Finally, the noise information only depends on the leading frames of an utterance, which may make the noise-corrupted speech codebook less accurate. This problem will be worse if the noise is non-stationary. Although updating the noise estimate within an utterance based on a voice activity detection (VAD) process can alleviate this problem, it will substantially increase the implementation complexity.

Based on the above observations, in this paper, we propose to improve the accuracy of the feature statistics estimation in two aspects. First, the procedures of creating the pseudo stereo codebooks are modified so that they are more representative of the speech features. The resulting advanced pseudo stereo codebooks are shown to be more effective in the codebook-based methods than the original ones. Second, the information from both the codebook and the frames of the processed utterance are integrated, so that more accurate statistics of the features can be obtained in order to further enhance the feature statistics normalization techniques. This idea is realized on three well-known approaches, CMS, CMVN, and HEQ. We will show that the resulting "associative" methods are superior to the original

utterance-based and codebook-based ones in the Aurora-2 clean-condition training task.

The remainder of the paper is organized as follows: Section 2 presents the construction of advanced pseudo stereo codebooks. Section 3 introduces our proposed associative cepstral normalization techniques. The experimental environment setup is described in Section 4, and the recognition results are given and discussed in Section 5. Finally, Section 6 contains brief conclusions.

2. The Construction of Advanced Pseudo Stereo Codebooks

In this section, we introduce the approach to constructing the advanced pseudo stereo codebooks for clean training and noise-corrupted testing environments, respectively. The corresponding procedures are also shown in Figure 1. The basic idea of the process for constructing these codebooks is as follows: during the feature extraction processes, we find an intermediate feature domain in which the clean speech and noise are *linearly additive* (assuming that the speech signal and noise are uncorrelated in the time domain). The clean speech codewords for the intermediate feature domain first are constructed then are linearly added to the noise estimates to compose the noise-corrupted speech codewords for that domain. Finally, they are transformed to the final feature domain following the remaining feature extraction processes. For the mel-frequency cepstral coefficients (MFCC), the intermediate feature mentioned above is the mel-spectrum, while for the other two types of speech features, linear prediction cepstral coefficients (LPCC) (Atal, 1974; Makhoul, 1975) and perceptual linear prediction cepstral coefficients (PLPCC) (Hermansky, 1990), both the auto-correlation coefficients and the magnitude spectrum can be selected as the intermediate feature. Therefore, the codebook construction process and the relating methods can be applied to MFCC, LPCC, and PLPCC.

For simplicity, the mel-frequency cepstral coefficients (MFCC) are used as the speech features here; thus, the two codebooks are just designed for MFCCs. Following the derivation processes of the MFCCs for a speech signal, the speech portions of all clean speech utterances in the training database are converted into sequences of mel-spectral vectors, each consisting of the mel-filter bank outputs. These vectors are then used to construct a set of R codewords together with their weights by vector quantization (VQ), denoted as:

$$\left\{\tilde{\mathbf{x}}[r], w_r;\ 1 \le r \le R\right\}, \tag{1}$$

where $\tilde{\mathbf{x}}[r]$ and w_r represent the r^{th} codeword and its corresponding weight, respectively. Each weight w_r represents the relative cluster size in VQ classification. These mel-spectral codewords are then transformed into the cepstral domain as follows:

$$\mathbf{x}[r] = \mathbf{C}\log\left(\tilde{\mathbf{x}}[r]\right), \tag{2}$$

where $\mathbf{C}$ is the discrete-cosine-transform (DCT) matrix. Thus, the set of codewords $\left\{\tilde{\mathbf{x}}[r], w_r;\ 1 \le r \le R\right\}$ is the clean speech cepstral codebook. Note that we construct this cepstral codebook by transforming the mel-spectral codewords rather than by vector quantizing the cepstral features directly and that these mel-spectral codewords are preserved in order to construct the noise-corrupted speech codebook.

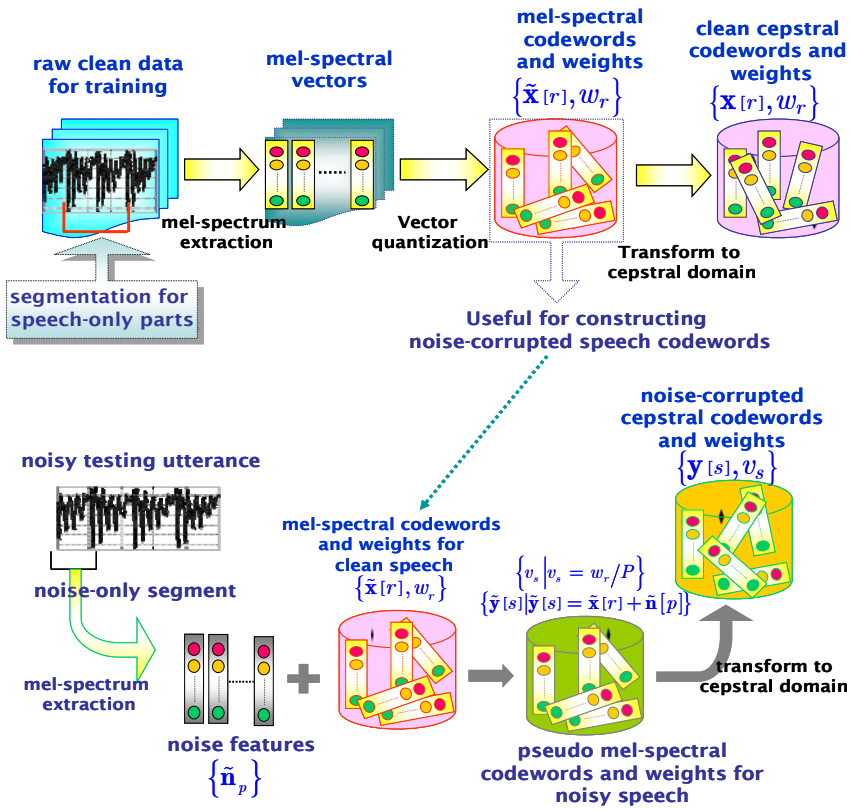

Figure 1. The procedures of constructing advanced pseudo stereo codebooks.

For the noise-corrupted testing environment, since it is often difficult to obtain a set of reliable codewords completely based on a single testing utterance, we construct the "noise-corrupted speech" codebook with the help of the available "clean-speech" mel-spectral codewords. For a given noise-corrupted testing utterance, let the estimated mel-spectra of the noise be approximated as a set of vectors, which are denoted as $\left\{\tilde{\mathbf{n}}[p];1 \le p \le P\right\}$, where P is the number of noise vectors. Then, since the clean speech and noise are approximately additive in the mel-spectral domain, the mel-spectral codewords for the noise-corrupted utterance are obtained as

$$\tilde{\mathbf{y}}[s]\Big|_{s=(r-1)P+p} = \tilde{\mathbf{x}}[r] + \tilde{\mathbf{n}}[p]. \tag{3}$$

and the weight for each $\tilde{\mathbf{y}}[s]$ is approximated by

$$v_s\Big|_{s=(r-1)P+p} = w_r / P\,. \tag{4}$$

Finally, we transform each $\tilde{\mathbf{y}}[s]$ into the cepstral domain, as in Eq. (2):

$$\mathbf{y}[s] = \mathbf{C}\log\left(\tilde{\mathbf{y}}[s]\right), \tag{5}$$

where $\mathbf{C}$ is the discrete-cosine-transform (DCT) matrix. Thus, $\left\{\mathbf{y}[s], v_s; 1 \le s \le RP\right\}$ is the noise-corrupted speech cepstral codebook. From the above, the two sets of codewords, $\left\{\mathbf{x}[r], w_r\right\}$ and $\left\{\mathbf{y}[s], v_s\right\}$, are viewed as the representatives for the clean training and noise-corrupted testing conditions, respectively, and they are named "pseudo stereo codebooks" here. The term "pseudo" indicates that the noise-corrupted speech codebook is *not* derived from the noise-corrupted speech directly, but is a fusion of the clean speech codebook and the noise estimates.

Here, the codebook construction scheme is different from that in (Hilger & Ney, 2006) in two points:

1. A speech/non-speech classification, or voice activity detection (VAD) procedure, is performed on each utterance in the clean training database, and then *only the speech portions* of these utterances are used to construct the clean-speech codewords. Thus, the resulting codewords will convey the speech characteristics better than those obtained in (Hilger & Ney, 2006). More precisely, for almost every utterance in the clean training set, there is a relatively long silent portion preceding and/or following the speech-containing portion. Therefore, if we create the codewords with utterances that are not VAD-processed, there may be a significant number of codewords that correspond to the silence, or the codewords corresponding to the silence may possess relatively high weight values, which will result in a less accurate estimate for the statistics of speech features. Furthermore, since the utterances to be VAD-processed here are clean from noise, the results of speech/non-speech classification are very accurate.
2. Second, the codeword weights, $\left\{w_r\right\}$ and $\left\{v_s\right\}$, are additionally calculated. They represent the relative significance of each codeword; thus, the estimated statistics of the speech features based on the advanced codebooks are expected to be more accurate than those on the original ones in (Hung, 2008).

In the above codebook construction process, we only focus on the speech characteristics in an utterance for clean and noise-corrupted conditions, while the non-speech portions

(silence or noise-only regions) are not considered. In other words, a "non-speech" codebook is not constructed here. The main reason is simple: in the feature statistics normalization approaches, we often do not process the speech and non-speech frames separately in an utterance, but treat them in the same manner as the same estimated feature statistics. Although normalizing the speech and non-speech frames based on different feature statistics may bring better recognition performance, it requires a reliable voice activity detector (VAD) which classifies a frame as speech or non-speech accurately in both clean and noise-corrupted conditions. Nevertheless, in general, a VAD brings about higher mis-classification rates as the signal-to-noise ratio (SNR) gets worse in the noise-corrupted condition, thus possibly harming the performance of feature statistics normalization approaches.

With the help of the above advanced pseudo stereo codebooks, we can estimate the statistics and probability distribution functions for the features of both the clean and noise-corrupted speech. For example, given the time stream of the m^{th} cepstral coefficients, $\{c_m[n]\}$, of *a clean utterance in the training set*, the k^{th} order moment and the probability distribution function of C_m are approximated by

$$E\left\{\left(C_m\right)^k\right\} \approx \sum_{r=1}^{R} w_r \left(x_m[r]\right)^k, \quad k = 1,2,3,\cdots, \tag{6}$$

and

$$F_{C_m}\left(z\right) \triangleq P\left(C_m \le z\right) \approx \sum_{r=1}^{R} w_r u\left(z - x_m[r]\right), \tag{7}$$

respectively, where C_m denotes a random variable with the samples $\{c_m[n]\}$, $x_m[r]$ is the m^{th} component of the clean speech codeword $\mathbf{x}[r]$, and $u(.)$ is the unit step function, defined by

$$u\left(\ell\right) = \begin{cases} 1 & \text{if } \ell \ge 0 \\ 0 & \text{if } \ell < 0 \end{cases}. \tag{8}$$

Similarly, if the time stream $\{c_m[n]\}$ corresponds to *a noise-corrupted utterance in the testing set*, then the k^{th} order moment and the probability distribution function of C_m are approximated by:

$$E\left\{\left(C_m\right)^k\right\} \approx \sum_{s=1}^{RP} v_s \left(y_m[s]\right)^k, \quad k = 1,2,3,\cdots, \tag{9}$$

and

$$F_{C_m}\left(z\right) \triangleq P\left(C_m \le z\right) \approx \sum_{s=1}^{RP} v_s u\left(z - y_m[s]\right) \tag{10}$$

respectively, where $y_m[s]$ is the m^{th} component of the noise-corrupted speech codeword $\mathbf{y}[s]$.

Based on these estimated statistics from the advanced codebooks, as in Eqs. (6), (7), (9), and (10), the codebook-based cepstral statistics normalization methods stated in (Hung, 2008) can be implemented. In Section 5, we will show that the advanced codebooks benefit the codebook-based CMS and CMVN in improving the recognition accuracy under noise-corrupted environments.

3. Associative Cepstral Statistics Normalization Techniques

The previous section introduces how to construct a better codebook set in order to enhance the corresponding codebook-based feature normalization methods. Updating the noise information in the noise-corrupted codebook, however, especially for a non-stationary noise environment, still makes the normalization method less efficient in computation. Besides, the codebook-based methods do not behave very well for some normalization methods, like HEQ, which will be shown in the subsequent sections. As a result, we attempt to incorporate the whole-utterance frames with the developed codebooks to evaluate the feature statistics, in the hope that the resulting feature statistics normalization techniques can bring better recognition accuracy. We realize our idea in the methods of CMS, CMVN, and HEQ, respectively, which is described in the following three subsections.

3.1 Associative Cepstral Mean Subtraction

Cepstral mean subtraction (CMS) (Atal, 1974) is a well-known speech feature processing technique. It was initially developed for eliminating the channel distortion in the features, but was found to be helpful as well in alleviating the effect of additive noise. In CMS, the original features are normalized to have zero mean. Briefly speaking, with the time-trajectory of the m^{th} cepstral coefficients, $\{c_m[n]\}$, for an utterance as the input, the output of the CMS process is expressed as:

$$\tilde{c}_m[n] = c_m[n] - \mu_m, \quad 1 \le n \le N, \tag{11}$$

where N is the number of frames in the utterance, and μ_m is the mean (the first-order moment) of $c_m[n]$. Here, the parameter μ_m is estimated by incorporating the codebooks, $\{\mathbf{x}[r], w_r\}$ and $\{\mathbf{y}[s], v_s\}$, in Section 2 and the whole-utterance frames, $\{c_m[n], 1 \le n \le N\}$. That is, for a clean speech utterance in the training set,

$$\mu_m = \alpha\left(\sum_{r=1}^{R} w_r x_m[r]\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} c_m[n]\right), \tag{12}$$

and for a noise-corrupted speech utterance in the testing set,

$$\mu_m = \alpha\left(\sum_{s=1}^{RP} v_s y_m[s]\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} c_m[n]\right). \tag{13}$$

In Equations (12) and (13), $x_m[r]$ and $y_m[s]$ denote the m^{th} component of the codewords $\mathbf{x}[r]$ and $\mathbf{y}[s]$, respectively, and α is a weighting factor between 0 and 1, which determines the usage ratio between the codebook and the whole-utterance frames. Here, the CMS method with the mean parameters defined in Eqs. (12) and (13) is named associative CMS (A-CMS). Obviously, if α is set to 1, the information from the frames in the utterance is completely ignored, and A-CMS is identical to codebook-based CMS (C-CMS). On the other hand, A-CMS with $\alpha = 0$ behaves equally to utterance-based CMS (U-CMS).

3.2 Associative Cepstral Mean and Variance Normalization

In the method of cepstral mean and variance normalization (CMVN) (Tibrewala & Hemansky, 1997), the original features are normalized to have zero mean and unity variance. With the time-trajectory of the m^{th} cepstral coefficients, $\{c_m[n]\}$, for an utterance as the input, the output of the CMVN process is expressed as:

$$\tilde{c}_m[n] = \left(c_m[n] - \mu_m\right)\Big/\sigma_m, \quad 1 \le n \le N, \tag{14}$$

where N is the number of frames in the utterance, while μ_m and σ_m are the mean and standard deviation of $c_m[n]$, respectively. In general, CMVN performs better than CMS because it additionally normalizes the variance of the features.

Similar to the previous sub-section, we estimate the two parameters, μ_m and σ_m, by incorporating the codebooks and the whole-utterance frames. That is, for a clean speech utterance in the training set,

$$\mu_m = \alpha\left(\sum_{r=1}^{R} w_r x_m[r]\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} c_m[n]\right), \tag{15}$$

$$\sigma_m^2 = \alpha\left(\sum_{r=1}^{R} w_r x_m^2[r]\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} c_m^2[n]\right) - \mu_m^2, \tag{16}$$

and for a noise-corrupted speech utterance in the testing set,

$$\mu_m = \alpha\left(\sum_{s=1}^{RP} v_s y_m[s]\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} c_m[n]\right), \tag{17}$$

$$\sigma_m^2 = \alpha\left(\sum_{s=1}^{RP} v_s y_m^2[s]\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} c_m^2[n]\right) - \mu_m^2. \tag{18}$$

In Equations (15)-(18), $x_m[r]$ and $y_m[s]$ denote the m^{th} component of the codewords $\mathbf{x}[r]$ and $\mathbf{y}[s]$, respectively, and α is a weighting factor between 0 and 1, which

determines the usage ratio between the codebook and the whole-utterance frames. Here, the CMVN method with means and variances defined in Eqs. (15)-(18) is named associative CMVN (A-CMVN). Similar to the case in the previous subsection, A-CMVN with $\alpha = 1$ is equivalent to codebook-based CMVN (C-CMVN), while A-CMVN with $\alpha = 0$ behaves equally to utterance-based CMVN (U-CMVN).

3.3 Associative Histogram Equalization

The histogram equalization (HEQ) technique (Hsu & Lee, 2004) normalizes each cepstral component stream so that the resulting histogram is close to a reference function. Following the notation of the previous two subsections, with the time-trajectory of the m^{th} cepstral coefficients $\left\{c_m[n]\right\}$ for an utterance as the input, the output of the HEQ process can be expressed as:

$$\tilde{c}_m[n] = F_N^{-1}\left(F_{C_m}\left(c_m[n]\right)\right), \quad 1 \le n \le N, \tag{19}$$

where $F_{C_m}(.)$ is the probability distribution function of $\left\{c_m[n]\right\}$, and $F_N(.)$ is a pre-defined reference distribution function. Compared with CMS and CMVN, HEQ additionally compensates all the higher-order moments of the features, and this extra compensation often results in an apparent improvement.

Analogous to the previous subsections, the distribution function $F_{C_m}(.)$ is jointly determined by the codebooks and the whole-utterance frames, and the resulting algorithm is called associative HEQ (A-HEQ). In A-HEQ, for a clean speech utterance in the training set,

$$F_{C_m}(z) = \alpha\left(\sum_{r=1}^{R} w_r u\left(z - x_m[r]\right)\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} u\left(z - c_m[n]\right)\right), \tag{20}$$

and for a noise-corrupted speech utterance in the testing set,

$$F_{C_m}(z) = \alpha\left(\sum_{s=1}^{RP} v_s u\left(z - y_m[s]\right)\right) + (1-\alpha)\left(\frac{1}{N}\sum_{n=1}^{N} u\left(z - c_m[n]\right)\right), \tag{21}$$

where $u(.)$ is the unit step function, as in Eq. (8).

Again, in Equations (20) and (21), the weighting factor α determines the usage ratio between the codebook and the whole-utterance frames. In the extreme case, $\alpha = 1$, the distribution function is completely determined by the codebook, thus A-HEQ becomes codebook-based HEQ (C-HEQ). In the other extreme case of $\alpha = 0$, A-HEQ corresponds to utterance-based HEQ (U-HEQ).

3.4 Comparison with Some Other Noise Compensation Algorithms

Previous work presents a series of noise compensation approaches which consider the speech and noise characteristics simultaneously, including the parallel model combination (PMC) (Gales & Young, 1993; 1995a; 1995b), vector Taylor series (VTS) (Acero, Deng, Kristjansson, & Zhang, 2000; Segura, Benitez, de la Torre, Dupont, & Rubio, 2002; Moreno, Raj, & Stem, 1998), and stereo-based piecewise linear compensation for environments (SPLICE) (Deng, Acero, Jiang, Droppo, & Huang, 2001; Droppo, Deng, & Acero, 2001). Here, we discuss the relationship of our proposed methods with PMC, VTS, and SPLICE, as well as the differences among them as follows:

1. In PMC, the original clean speech model parameters in the cepstral domain are transformed to the linear spectral domain, combined with the noise model parameters, then transformed back to the cepstral domain to be the approximated noisy speech model. Therefore, PMC compensates the speech model while keeping the noisy testing speech unchanged. Similar to PMC, in our proposed methods, the noisy speech codewords are obtained by integrating the clean speech codewords and the noise estimates in the linear spectral domain. Nevertheless, in our method, both the clean training and noisy testing speech data are compensated, then the speech model is trained (not just modified) with the new clean training speech data.
2. The VTS algorithm is often applied in two directions: one to compensate the speech model while keep the noisy testing speech unchanged, and the other to compensate the noisy testing speech without altering the original clean speech model. Briefly speaking, VTS considers that noisy speech is a nonlinear function of clean speech and noise in the logarithmic spectral domain and that this nonlinear function is approximated as a polynomial in order to estimate the statistics of noisy speech with the statistics of clean speech of noise. Therefore, our proposed methods differ from VTS in two ways: both the noisy testing speech and the speech model are changed in our methods and we primarily deal with the relationship of clean speech and noise in the linear spectral domain.
3. In SPLICE, the restored clean speech cepstral vector is obtained by adding the noisy speech cepstral vector to a correction vector. The correction vector is trained using the stereo recordings for both the clean and noisy speech data based on the maximum likelihood principle. In fact, in SPLICE, a Gaussian mixture model (GMM) for noisy speech cepstral vectors is trained, and the minimum-mean-square-error (MMSE) rule or the approximate maximum *a posteriori* (MAP) rule is applied to obtain the optimal estimate of the clean speech cepstral vector, given the noisy speech cepstral vector. Therefore, compared with SPLICE, our proposed methods do not use the stereo data since the noisy speech codebook is constructed simply by integrating the clean speech

codewords and the noise estimates. In addition, in SPLICE, the VQ process is performed on the noisy speech data in the cepstral domain, while in our methods we implement the VQ process on the clean speech data in the linear spectral domain.

To sum up briefly, in PMC, VTS, and SPLICE, the clean speech or noisy speech is often modeled by a single Gaussian or a mixture of Gaussians, while our proposed methods the speech are partially represented by a set of codewords. Furthermore, our proposed methods have lower computation complexity than PMC, VTS, and SPLICE, while they can provide very good recognition performance, as will be shown in the next section.

4. Experimental Setup

The proposed codebook-based algorithms have been tested with the AURORA-Project Digit Database Version 2.0, which is described in detail in (Hirsch & Pearce, 2000). In this database, the recordings have been manually segmented into utterances, and each utterance is saved as a file. The number of digits in an utterance can be one, two, three, four, five, six, and seven. Besides the digits, there is always a silent section at the beginning and end of an utterance. The length of an utterance may vary from 0.59 sec to 5.15 sec, depending on the number of digits in the utterance. The testing data consist of 4004 utterances from 52 male and 52 female speakers. Three different subsets are defined: Test Set A and Test Set B are each affected by four types of noise, and Test Set C is affected by two types. The noises included are: subway, babble, car, exhibition, restaurant, street, airport, and train station. Each noise is added to the clean speech under seven different signal-to-noise ratios (SNRs): -5dB to 20dB, spaced in 5dB intervals, and clean (no noise). The signals in Set A and Set B are filtered with a G.712 filter, and those in Set C are filtered with a MIRS filter. G.712 and MIRS are two standard frequency characteristics defined by the ITU (ITU recommendation G.712, 1996). Since the proposed methods are focused on improving the recognition accuracy for an additive noise environment, only Set A and Set B are used for the subsequent experiments.

On the other hand, under the clean training condition, the training data consist of 8440 clean speech utterances produced by 55 male and 55 female adults. These signals are filtered with a G.712 filter without noise added. For the clean training phase, the 8440 strings in the training set are first processed by an energy-based VAD process (Tai & Hung, 2006), and the speech portions are converted into vector streams of 23 mel-spectral coefficients. All of the 23-dimensional feature vectors are used to construct a set of R codewords via vector quantization (VQ) with the *K*-means clustering algorithm, in which the squared Euclidean distance is used for VQ classification. These codewords are also converted to 13-dimensional mel-frequency cepstral vectors (c_0~c_{12}) to form the clean speech cepstral codebook $\left\{\mathbf{x}[r], w_r; 1 \le r \le R\right\}$. Besides, all 8440 strings (including speech and non-speech portions) in the training set are converted to MFCC feature vector streams. The resulting 13-dimensional

cesptral features plus their delta and delta-delta comprise the components of the final 39-dimensional feature vectors. With these feature vectors in the training set, two sets of hidden Markov models (HMMs) for each digit (oh, zero, one, …, eight, and nine) and silence are trained. The first set follows the Microsoft complex back-end training scripts (Droppo, Deng, & Acero, 2002), in which each digit HMM has 16 states and 20 Gaussian mixtures per state. The second set follows the standard training scripts provided in the Aurora-2 database (Hirsch & Pearce, 2000), in which each digit HMM has 16 states and 3 Gaussian mixtures per state.

For the testing phase, the leading 10 frames (0.1 sec) of each utterance are assumed to be noise-only, and their corresponding 23-dimensional mel-spectral vectors are the elements of the estimated noise components $\left\{\tilde{\mathbf{n}}[p];\ 1 \le p \le P\right\}$ (where $P = 10$). We then construct the noise-corrupted speech cepstral codebook $\left\{\mathbf{y}[s], v_s; 1 \le s \le RP\right\}$ following the procedures in Section 2. Based on the two codebooks $\left\{\mathbf{x}[r], w_r\right\}$ and $\left\{\mathbf{y}[s], v_s\right\}$, the proposed algorithms are performed to adjust the features for both training and testing. The reference distribution function in Equation (19) for HEQ is a Gaussian distribution with zero mean and unity variance. Note that, even though it has been shown that dynamically determining the noise-only components within an utterance based on a voice activity detector (VAD) improves the recognition performance of codebook-based methods (Hung, 2008), it will significantly increase the computation complexity, especially when the detected non-speech portion is quite long (the size of noise-corrupted speech codebook, RP, is proportional to the number of noise components, P). Besides, the results of VAD become less reliable when the signal-to-noise ratio (SNR) gets worse, which somewhat deteriorates the accuracy of the resulting noise-corrupted speech codebook. Based on the above observations, we select the first 0.1 second (10 frames) of each utterance as the noise-only components, in which there is always no speech.

5. Experimental Results and Discussions

We compare and analyze the recognition accuracy achieved by the different approaches proposed here for the AURORA-2 experimental environment. This includes 5 subsections. In Subsections 5.1-5.4, the experimental results are obtained via the first set of HMMs (the complex back-end). Subsection 5.5, on the other hand, presents the experimental results obtained via the second set of HMMs (the standard back-end). The results in the first four subsections help us investigate the best possible recognition accuracy achieved by the proposed methods with a more elaborate model structure, while the results in the last subsection can be used with the purpose of performance comparison with many other robustness techniques which are evaluated under the standard model structure.

Briefly speaking, in Subsection 5.1 we examine the advanced pseudo stereo codebooks

proposed in Section 2 to see if they bring better recognition results in codebook-based CMS and CMVN than the original ones proposed in (Hung, 2008). In Subsection 5.2, the proposed associative CMS, CMVN, and HEQ in Section 3 are compared with the corresponding utterance-based and codebook-based ones in terms of the recognition performance. In Subsection 5.3, the effect of the weighting factor α in associative approaches of Section 3 is analyzed to see the corresponding influence on the recognition performance. In Subsection 5.4, we compare the proposed approaches with some other noise-robust techniques. Finally, the proposed methods are evaluated with the standard back-end in Subsection 5.5.

5.1 Comparison of the Advanced Pseudo Stereo Codebooks with the Original Ones in Codebook-based Approaches

We compare the new proposed pseudo stereo codebooks in Section 2 with the original ones in (Hung, 2008) in terms of the recognition accuracy for the codebook-based CMS and CMVN. For the pseudo stereo codebooks, the number of clean speech codewords, R, is set to 16, 64, and 256, and the first 10 frames of each utterance are assumed to be noise-only and their corresponding features constitute the estimated noise vectors $\left\{\tilde{\mathbf{n}}[p],\ 1 \le p \le P = 10\right\}$. Thus, the number of noise-corrupted speech codewords, RP, is equal to 160, 640, or 2560.

Tables 1 and 2 list the recognition results of codebook-based CMS and CMVN, respectively, where these results present individual set recognition accuracy rates averaged over five SNR conditions (0~20dB, at 5dB intervals). For the purpose of clarification in the tables, a superscript "$^{(o)}$" is added to the names of C-CMS and C-CMVN to indicate that the two methods are based on the original pseudo stereo codebooks.

From the two tables, several phenomena can be found:

1. Both CMS and CMVN bring improvement in recognition accuracy when compared with the baseline processing. As expected, however, CMVN performs better than CMS in all cases. As a result, performing the variance normalization really helps improve the noise robustness of speech features.

2. Under the same assignment for the parameter R (the number of clean speech codewords), the advanced pseudo stereo codebooks provide C-CMS and C-CMVN with significantly better recognition accuracy than the original codebooks do. These results support our statement in Section 2 that the proposed new codebooks are capable of providing a better estimate of feature statistics, thus, benefit the codebook-based approaches.

3. In most cases, increasing the number of clean speech codewords R brings about improved recognition accuracy for the four methods, C-CMS$^{(o)}$, C-CMS, C-CMVN$^{(o)}$, and C-CMVN. For both C-CMS and C-CMVN with the advanced codebooks, however, a

moderate number of codewords already give rise to nearly optimal performance. Nevertheless, this is not the case for C-CMS$^{(o)}$ and C-CMVN$^{(o)}$, with the possible reason that, in these two methods, we have to increase the number of codewords so that more of them can represent the speech portions, and more accurate feature statistics can be estimated accordingly. As a result, it shows that the advanced codebooks with a moderate size can serve as good representatives for speech features.

Table 1. Recognition accuracy (%) achieved by two versions of codebook-based CMS with a different number of clean speech codewords R averaged across the SNRs between 0 and 20dB, under the complex back-end structure. AR (%) and RR (%) are the absolute and relative error rate reductions over the baseline. C-CMS uses the advanced pseudo stereo codebooks, while C-CMS$^{(o)}$ uses the original pseudo stereo codebooks in (Hung, 2008).

Method	Set A	Set B	Average	AR	RR
Baseline	71.92	67.79	69.86	—	—
C-CMS$^{(o)}$ ($R = 16$)	74.21	70.81	72.51	2.65	8.81
C-CMS ($R = 16$)	79.04	79.56	79.30	9.45	31.33
C-CMS$^{(o)}$ ($R = 64$)	74.03	70.74	72.39	2.53	8.39
C-CMS ($R = 64$)	80.79	80.19	80.49	10.64	35.28
C-CMS$^{(o)}$ ($R = 256$)	77.92	75.20	76.56	6.71	22.24
C-CMS ($R = 256$)	81.46	81.49	81.48	11.62	38.55

Table 2. Recognition accuracy (%) achieved by two versions of codebook-based CMVN with a different number of clean speech codewords R averaged across the SNRs between 0 and 20dB, under the complex back-end structure. AR (%) and RR (%) are the absolute and relative error rate reductions over the baseline. C-CMVN uses the advanced pseudo stereo codebooks, while C-CMVN$^{(o)}$ uses the original pseudo stereo codebooks in (Hung, 2008).

Method	Set A	Set B	average	AR	RR
Baseline	71.92	67.79	69.86	—	—
C-CMVN$^{(o)}$ ($R = 16$)	84.44	82.40	83.42	13.57	45.00
C-CMVN ($R = 16$)	85.41	85.21	85.31	15.46	51.27
C-CMVN$^{(o)}$ ($R = 64$)	84.13	81.53	82.83	12.98	43.04
C-CMVN ($R = 64$)	86.92	86.81	86.87	17.01	56.43
C-CMVN$^{(o)}$ ($R = 256$)	86.67	86.25	86.46	16.61	55.08
C-CMVN ($R = 256$)	87.10	87.32	87.21	17.36	57.57

5.2 Comparison of the Associative CMS, CMVN and HEQ with the Utterance-based and Codebook-based Approaches

The proposed associative cepstral normalization methods (A-CMS, A-CMVN, and A-HEQ) are evaluated here in terms of their robustness against noise. For the purpose of comparison, the experiments for the corresponding utterance-based and codebook-based methods are also performed. Here, the weighting factor α in Equations (12), (13), (15)-(18), (20), and (21) is preliminarily set to 0.5. Similar to the previous subsection, the size of the clean speech codebook, *R*, is set to 16, 64, or 256, and the number of leading frames for noise estimation, *P*, is set to 10.

Tables 3, 4, and 5 list the recognition results of various types of CMS, CMVN, and HEQ, respectively, where these results present individual set recognition accuracy rates averaged over five SNR conditions (0~20dB, at 5dB intervals). For example, in Table 3, the recognition accuracy rates for utterance-based CMS (U-CMS), codebook-based CMS (C-CMS) with $R = 256$, and associative CMS (A-CMS) with three assignments of the parameter R, are presented. A similar arrangement holds for Tables 4 and 5. From the three tables, a series of observations are obtained as follows:

1. Among the three types of CMS, A-CMS performs the best, followed by C-CMS and then U-CMS. This condition also holds for A-CMVN, C-CMVN, and U-CMVN. First, the results agree with those obtained in (Hung, 2008) that codebook-based CMS and CMVN behave better than utterance-based ones. Second, the associative CMS (A-CMS) and CMVN (A-CMVN) always outperform both their corresponding utterance-based and codebook-based ones. Therefore, combining codebooks with the processed utterance features in estimating the feature statistics indeed helps improve the recognition accuracy considerably.

2. For the three utterance-based methods, HEQ always performs better than CMVN and CMS. As stated in Section 3, compared with CMVN and CMS, HEQ additionally compensates for all the higher-order moments of features, thus, brings about extra improvement. This, however, is not the case for codebook-based methods: C-HEQ behaves worse than C-CMVN and is the worst of the three HEQ methods. A possible reason is that the codebooks give more accurate gross information (*i.e*. the mean and variance) for the features, but they are less capable of providing the detailed behavior (*i.e*. the probability distribution) of them.

3. Similar to the case in CMS and CMVN, the new A-HEQ outperforms U-HEQ and C-HEQ significantly. The superior performance of A-HEQ again reveals that the feature statistics can be estimated more accurately by incorporating the codebook and the processed utterance features.

4. In these high-performance A-CMS, A-CMVN, and A-HEQ, setting the weighting factor α to 0.5 implies the codebooks and the whole-utterance frames are equally treated without bias. Although $\alpha = 0.5$ is not necessarily an optimal assignment, at least it implies that there is little need for meticulous tuning of the weighting factor α in order to obtain an improved performance for these associative methods.

5. A particular phenomenon for these associative methods (A-CMS, A-CMVN, and A-HEQ) is that increasing the number of clean-speech codewords R does not improve the recognition accuracy, which somewhat contradicts the results for codebook-based CMS and CMVN, as shown in Tables 1 and 2. For example, increasing the value of R from 16 to 256 in A-HEQ results in an accuracy degradation of 1.40%. One of the possible reasons for this degradation is the inconsistency between the codebooks and the whole-utterance frames in these associative methods. The codebooks present only the characteristics of the speech portions in the utterance, while the whole utterance contains both speech and non-speech portions. Increasing the codebook size somewhat portrays the speech characteristics more precisely, thus, highlights the above inconsistency further. From the viewpoint of implementation, however, it becomes an advantage since we can obtain better recognition results with fewer codewords in these associative methods, which reduces the computation complexity.

Table 3. Recognition accuracy (%) achieved by utterance-based, codebook-based, and associative CMS averaged across the SNRs between 0 and 20dB, under the complex back-end structure. AR (%) and RR (%) are the absolute and relative error rate reductions over the baseline.

Method	Set A	Set B	Average	AR	RR
Baseline	71.92	67.79	69.86	—	—
U-CMS	79.37	82.47	80.92	11.07	36.71
C-CMS ($R = 256$)	81.46	81.49	81.48	11.62	38.55
A-CMS ($R = 16, \alpha = 0.5$)	83.28	84.92	84.10	14.25	47.25
A-CMS ($R = 64, \alpha = 0.5$)	82.92	84.89	83.91	14.05	46.61
A-CMS ($R = 256, \alpha = 0.5$)	82.13	84.29	83.21	13.36	44.30

Table 4. Recognition accuracy (%) achieved by utterance-based, codebook-based, and associative CMVN averaged across the SNRs between 0 and 20dB, under the complex back-end structure. AR (%) and RR (%) are the absolute and relative error rate reductions over the baseline.

Method	Set A	Set B	Average	AR	RR
Baseline	71.92	67.79	69.86	—	—
U-CMVN	85.03	85.56	85.30	15.44	51.22
C-CMVN ($R = 256$)	87.10	87.32	87.21	17.36	57.57
A-CMVN ($R = 16, \alpha = 0.5$)	87.87	88.67	88.27	18.42	61.09
A-CMVN ($R = 64, \alpha = 0.5$)	87.34	88.24	87.79	17.94	59.50
A-CMVN ($R = 256, \alpha = 0.5$)	87.25	88.06	87.66	17.80	59.05

Table 5. Recognition accuracy (%) achieved by utterance-based, codebook-based, and associative HEQ averaged across the SNRs between 0 and 20dB, under the complex back-end structure. AR (%) and RR (%) are the absolute and relative error rate reductions over the baseline.

Method	Set A	Set B	Average	AR	RR
Baseline	71.92	67.79	69.86	—	—
U-HEQ	86.95	88.39	87.67	17.82	59.10
C-HEQ ($R = 256$)	86.23	85.77	86.00	16.15	53.56
A-HEQ ($R = 16, \alpha = 0.5$)	90.21	91.16	90.69	20.83	69.10
A-HEQ ($R = 64, \alpha = 0.5$)	88.93	89.68	89.31	19.45	64.52
A-HEQ ($R = 256, \alpha = 0.5$)	88.84	89.73	89.29	19.43	64.46

Figures 2, 3, and 4 show the averaged recognition accuracy rates for each of the eight noise conditions in Test Sets A and B achieved by various types of CMS, CMVN, and HEQ. Roughly speaking, the four noise types, "subway," "street," "car," and "exhibition" can be viewed as stationary noise, while the other four noise types, "restaurant," "babble," "airport," and "train-station" are non-stationary noise. From the three figures, it is first found that, the utterance-based methods perform better in the non-stationary noise cases than in the stationary noise cases, while the situation is reversed for codebook-based methods. Second, the accuracy variation due to different noise conditions is more significant in the utterance-based and codebook-based methods than in the associative methods. Third, for each noise type, the associative method always performs better than the corresponding utterance-based and codebook-based ones, which again indicates that integrating the utterance and codebook information in these feature statistics methods is quite helpful for a wide range of noise

environments.

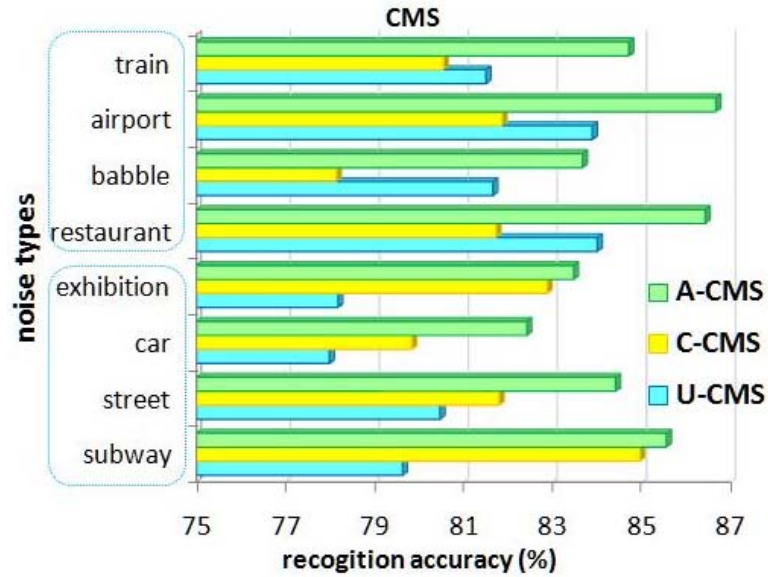

Figure 2. Recognition accuracy (%) achieved by three CMS methods, U-CMS, C-CMS (R=256), and A-CMS (R=16, α=0.5) for eight noise types in Test Sets A and B, averaged over five SNR conditions, 0~20dB.

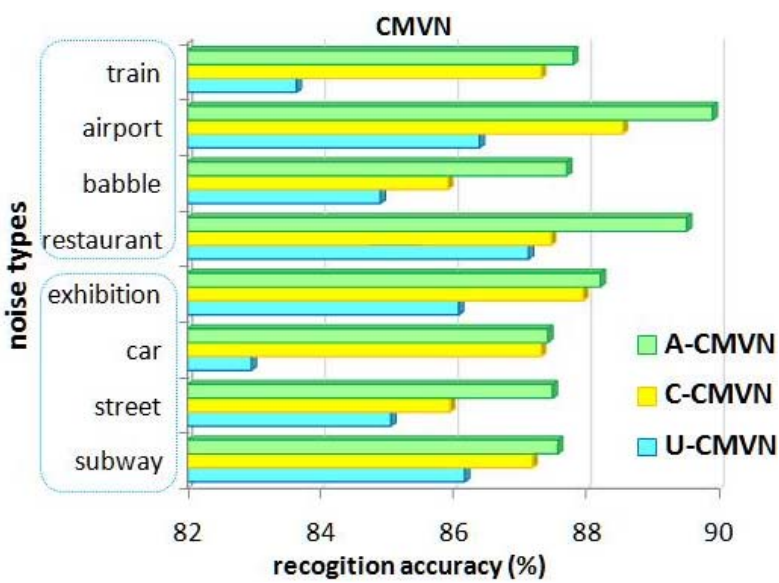

Figure 3. Recognition accuracy (%) achieved by three CMVN methods, U-CMVN, C-CMVN (R=256), and A-CMVN (R=16, α=0.5) for eight noise types in Test Sets A and B, averaged over five SNR conditions, 0~20dB.

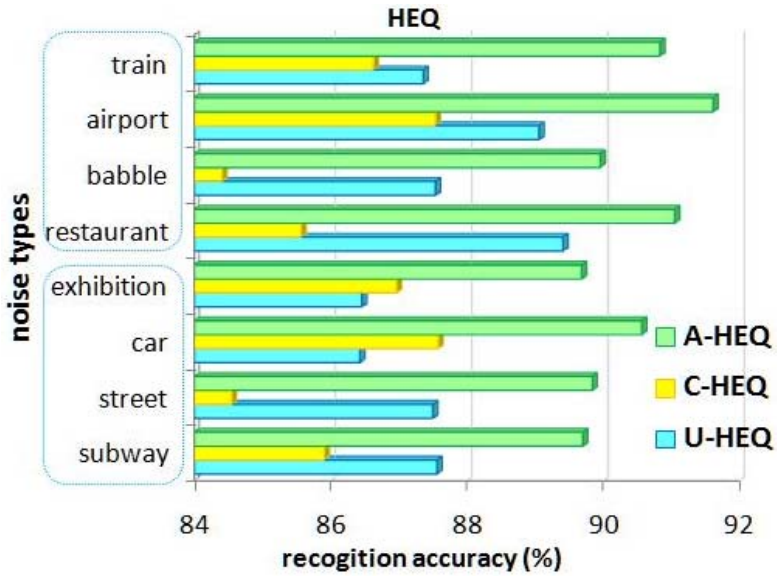

Figure 4. Recognition accuracy (%) achieved by three HEQ methods, U-HEQ, C-HEQ (R=256), and A-HEQ (R=16, α=0.5) for eight noise types in Test Sets A and B, averaged over five SNR conditions, 0~20dB.

5.3 The Effect of the Weighting Factor α in the Associative Methods

Here, the effect of the weighting factor α on the proposed associative methods is investigated. As stated in Section 2, the weighting factor α determines the usage ratio between the codebooks and the whole-utterance frames in estimating the feature statistics. Here, the size R of the clean speech codebook is fixed at 16 since it brings the best recognition accuracy, as mentioned in the previous subsection. Then, different assignments of the weighting factor α from 0 to 1, spaced at 0.1 intervals, are given for A-CMS, A-CMVN, and A-HEQ.

Figures 5, 6, and 7 show the recognition results averaged over five SNR conditions (0~20dB) and all eight noise types in Test Sets A and B for different values of α for A-CMS, A-CMVN, and A-HEQ, respectively. Note that these associative methods with $\alpha = 0$ and $\alpha = 1$ behave equally to the corresponding utterance-based and the codebook-based methods, respectively. For example, A-HEQ with $\alpha = 0$ is identical to U-HEQ, and A-HEQ with $\alpha = 1$ is identical to C-HEQ. From the three figures, we first find that, with any value of α that is not equal to 0 or 1, the newly-proposed associative methods always behave better than both the utterance-based and codebook-based ones. This result again supports our previous comment that integrating both codebook and utterance knowledge promotes the performance of feature statistics normalization techniques. Next, for different associative

methods, the corresponding optimal α values that achieve the optimal performance are not identical to each other. For example, the optimal α for A-CMS, A-CMVN, and A-HEQ are 0.3, 0.8, and 0.4, respectively. For each method, however, the differences among the accuracy rates obtained with different α are in fact relatively slight when α is in the range $[0.3, 0.8]$ (*i.e.*, $0.3 \le \alpha \le 0.8$). The maximum deviation in the accuracy rates for A-CMS with varying α is 1.69%, and it is 1.08% and 0.70% for A-CMVN and A-HEQ, respectively. Furthermore, setting $\alpha = 0.5$ just results in the accuracy degradation of 0.85%, 0.49%, and 0.01% for A-CMS, A-CMVN, and A-HEQ, respectively, when compared with the optimal values. This result implies that we just have to evenly employ the codebooks and the whole-utterance frames in these associative methods, then the nearly optimal performance can be achieved.

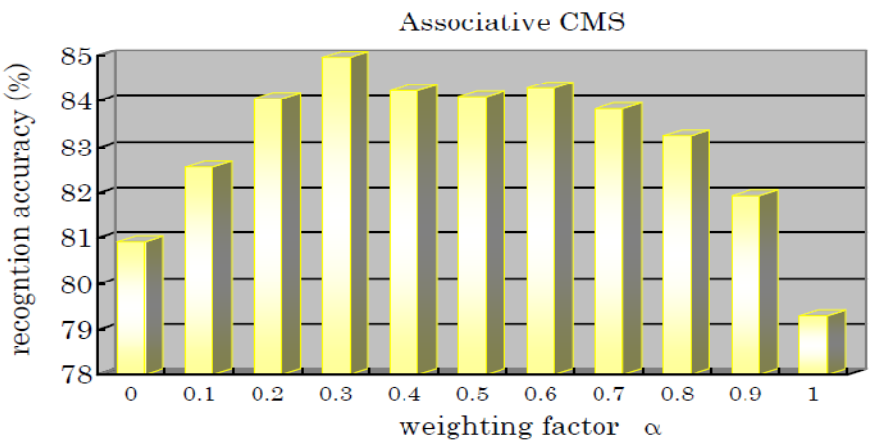

Figure 5. Recognition accuracy (%) averaged over five SNR values and all the eight noise types in Test Sets A and B vs. different assignments of the weighting factor α in associative CMS.

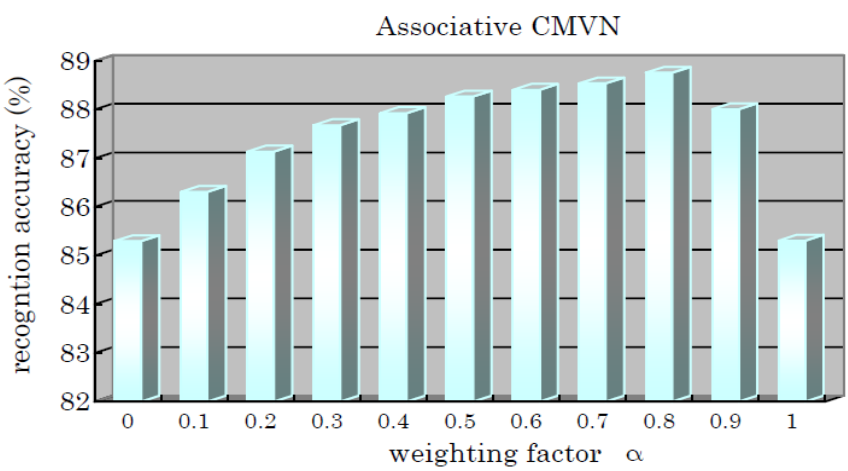

Figure 6. Recognition accuracy (%) averaged over five SNR values and all the eight noise types in Test Sets A and B vs. different assignments of the weighting factor α in associative CMVN.

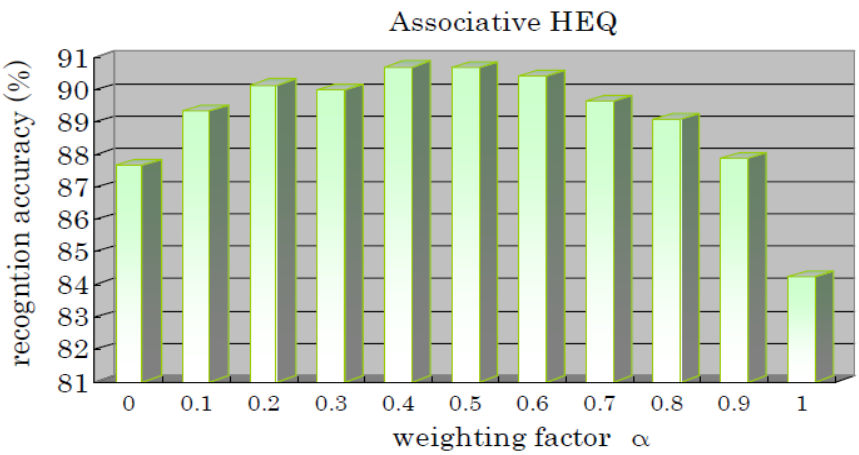

Figure 7. Recognition accuracy (%) averaged over five SNR values and all the eight noise types in Test Sets A and B vs. different assignments of the weighting factor α in associative HEQ.

5.4 Comparison of the Associative Methods with the Other Noise-Robust Techniques

In the previous subsections, we have shown that the associative methods outperform the corresponding utterance-based and codebook-based methods. Here, the associative methods are compared with the other two noise-robust techniques: mean-and-variance normalization plus autoregressive moving average (ARMA) filtering (MVA) (Chen & Bilmes, 2007) and the ETSI advanced front-end (AFE) feature extraction algorithm (ETSI standard doc, 2003). In MVA, an ARMA filter is performed on the (utterance-based) MVN-processed features in order to emphasize the relatively low modulation frequency components. On the other hand, the AFE makes use of a two-stage Wiener filter in order to reduce noise. For the purpose of comparison, we also perform the low-pass ARMA filtering on the A-CMVN processed features, which is called A-MVA here.

Figures 8 and 9 present the recognition accuracy rates of the various approaches for Test Sets A and B, respectively. From the two figures, we have the following observations:

1. Both MVA and AFE perform very well. The superior performance of MVA over U-CMVN shows that the low-pass ARMA filter helps extract the noise-robust components in U-CMVN processed features. On the other hand, AFE performs the best among all the methods here, which implies that, in AFE, the two-stage Wiener filtering process achieves very effective noise reduction.

2. A-CMVN behaves as well as MVA (U-CMVN plus ARMA), and in A-MVA the low-pass ARMA filtering process offers A-CMVN an accuracy improvement of 1.76% and

1.64% for Test Sets A and B, respectively. This result shows that, similar to U-CMVN, A-CMVN is additive to the ARMA filtering to provide better performance.

3. A-HEQ performs worse than AFE by 2.13% and 0.83% for Set A and Set B, respectively, in recognition accuracy. The possible explanation is: the noise estimates are quite accurate in AFE, which benefit the two-stage Wiener filtering process a lot, while A-HEQ just employs the several leading frames in an utterance as the noise estimates. As a result, in our future work, we will attempt to incorporate the noise estimates in AFE with the proposed associative methods in order to enhance their noise-robustness capability.

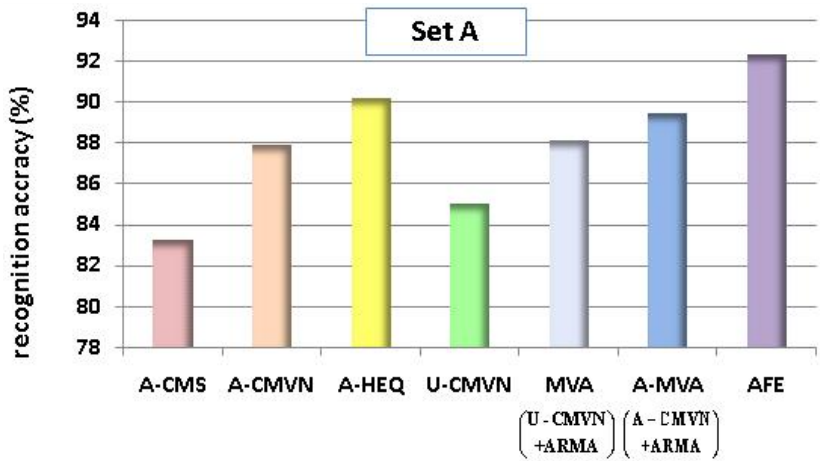

Figure 8. Recognition accuracy (%) achieved by various approaches averaged over five SNR values and all the four noise types in Test Set A, under the complex back-end structure.

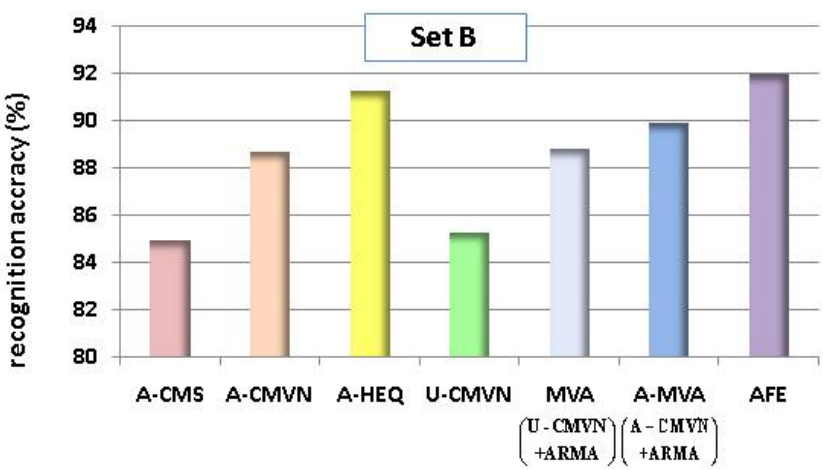

Figure 9. Recognition accuracy (%) achieved by various approaches averaged over five SNR values and all the four noise types in Test Set B, under the complex back-end structure.

5.5 Experimental Results of Our Proposed Methods with the Standard Back-end Defined in Aurora-2 Database

The recognition experiments in the previous four subsections use a more complicated hidden Markov model (HMM) structure, which follows the Microsoft advanced back-end training scripts (Droppo, Deng, & Acero, 2002), and each digit HMM has 16 states and 20 mixtures per states. Here, we train each digit HMM as 16 states and 3 mixtures per states following the standard back-end training scripts (Hirsch & Pearce, 2000), and the corresponding recognition results for our proposed methods are shown in Table 6. Comparing Table 6 with Tables 3, 4, and 5, we have the following observations:

1. Under the simpler HMM structure, the recognition accuracy rates achieved by each method become worse, which implies that a small number of mixtures cannot adequately represent the short-term speech characteristics.

2. The accuracy difference between the utterance-based and codebook-based methods becomes more significant. For example, C-CMS and C-CMVN outperform U-CMS and U-CMVN by 6.62% and, 6.71%, respectively (in Tables 3 and 4, the accuracy differences are less than 2%), and U-HEQ outperforms C-HEQ by 6.36% (in Table 5, the accuracy difference is just 1.67%).

3. In almost all cases, the proposed associative methods perform better than the corresponding utterance-based and codebook-based methods. Therefore, it reveals that, regardless of the recognition model complexity, integrating the codebook and utterance information helps enhancing the feature statistics normalization methods and thus brings about better recognition performance under additive noise environments.

Table 6. Recognition accuracy (%) achieved by various approaches averaged across the SNRs between 0 and 20dB, under the standard back-end structure. AR (%) and RR (%) are the absolute and relative error rate reductions over the baseline.

Method	Set A	Set B	Average	AR	RR
Baseline	61.97	55.78	58.88	—	—
U-CMS	64.36	69.43	66.90	8.02	19.50
C-CMS ($R = 256$)	72.40	74.64	73.52	14.64	35.60
A-CMS ($R = 16, \alpha = 0.5$)	73.72	77.51	75.61	16.73	40.69
U-CMVN	73.83	75.01	74.42	15.54	37.79
C-CMVN ($R = 256$)	80.88	81.39	81.13	22.25	54.11
A-CMVN ($R = 16, \alpha = 0.5$)	80.74	81.75	81.24	22.36	54.38
U-HEQ	80.33	81.24	80.79	21.91	53.28
C-HEQ ($R = 256$)	75.21	73.64	74.43	15.55	37.82
A-HEQ ($R = 16, \alpha = 0.5$)	82.96	83.85	83.41	24.53	59.65

6. Concluding Remarks and Future Works

In this paper, we propose associative CMS, CMVN, and HEQ, in which the required statistical information is obtained by incorporating advanced pseudo stereo codebooks and the processed whole-utterance frames. These new approaches demonstrate enhanced robustness of speech features under various additive noise environments. Compared with conventional utterance-based and codebook-based CMS, CMVN, and HEQ, these new approaches provide significantly better recognition performance. In our future work, we will employ the proposed statistics evaluation scheme to other feature statistics normalization approaches, like CGN (Yoshizawa, Hayasaka, Wada, & Miyanaga, 2004), HOCMN (Hsu & Lee, 2004), and CSN (Du & Wang, 2008), to investigate if better performance can be achieved.

References

Acero, A. (1990). *Acoustical and environmental robustness in automatic speech recognition*. Ph.D. dissertation, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburg, PA.

Acero, A., Deng, L., Kristjansson, T., & Zhang, J. (2000). HMM adaptation using vector Taylor series for noisy speech recognition. In *Proceeding of 2000 International Conference on Spoken Language Processing (ICSLP 2000)*, 3, 869-872.

Atal, B.S. (1974). Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *Journal of the Acoustic Society of America*, 55, 1304-1312.

Beattie, V. L., & Young, S. J. (1992). Hidden Markov model state-based cepstral noise compensation. In *Proceeding of International Conference on Spoken Language Processing (ICSLP 1992)*, 519-522.

Berstein, A. D., & Shallom, I. D. (1991). An hypothesized Wiener filtering approach to noisy speech recognition. In *Proceeding of 1991 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1991)*, 2, 913-916.

Boll, S.F. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 27(2), 113-120.

Chen, C.-P., & Bilmes, J. A. (2007). MVA Processing of Speech Features. *IEEE Trans on Audio, Speech, and Language Processing*, 15(1), 257-270.

Deng, L., Acero, A., Jiang, L., Droppo, J., & Huang, X. (2001). High-performance robust speech recognition using stereo training data. In *Proceeding of 2001 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, 1, 301-304.

Droppo, J., Deng, L., & Acero, A. (2001). Evaluation of the SPLICE Algorithm on the Aurora2 Database. In *Proceeding of 2001 Eurospeech Conference on Speech Communications and Technology (Eurospeech 2001)*, 217-220.

Droppo, J., Deng, L., & Acero, A. (2002). Evaluation of SPLICE on the AURORA 2 and 3 tasks. In *Proceeding of 2002 International Conference on Spoken Language Processing (Interspeech 2002－ICSLP)*, 29-32.

Du, J. & Wang, R.-H . (2008). Cepstral shape normalization (CSN) for robust speech recognition. In *Proceeding of 2008 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*, 4389-4392.

ETSI standard doc. (2003). Speech Processing, transmission and quality aspects (STQ); Distributed speech recognition; Advanced feature extraction algorithm. ETSI ES 202 050 v1.1.3 (2003-11).

Gales, M. J. F. & Young, S. J. (1993). Cepstral parameter compensation for HMM recognition in noise. *Speech Communication*, 12, 231-239.

Gales, M. J. F. & Young, S. J. (1995a). Robust speech recognition in additive and convolutional noise using parallel model combination. *Computer Speech and Language*, 9, 289-307.

Gales, M. J. F. & Young, S. J. (1995b). A fast and flexible implementation of parallel model combination. In *Proceeding of 1995 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1995)*, 133-136.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4), 1738-1752.

Hilger, F. & Ney, H. (2006). Quantile based histogram equalization for noise robust large vocabulary speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 14(3), 845-854.

Hirsch, H. G. & Pearce, D. (2000). The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In *Proceeding of ISCA ITRW ASR2000 "Automatic Speech Recognition: Challenges for the Next Millennium*", Paris, France, 181-188.

Holmes, J. N. & Sedgwick, N. C. (1986). Noise compensation for speech recognition using probabilistic models. In *Proceeding of 1986 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1986)*, 11, 741-744.

Hsu, C.-W. & Lee, L.-S. (2004). Higher order cepstral moment normalization (HOCMN) for robust speech recognition. In *Proceeding of 2004 International Conference on Acoustics, Speech and Signal Processing*, 197-200.

Hung, J.-W. (2006). Cepstral statistics compensation using online pseudo stereo codebooks for robust speech recognition in additive noise environments. In *Proceeding of 2006 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*.

Hung, J.-W. (2008). Cepstral statistics compensation and normalization using online pseudo stereo codebooks for robust speech recognition in additive noise environments. *IEICE Trans. on Information and Systems*, E91-D(2), 296-311.

ITU recommendation G.712. (1996). Transmission performance characteristics of pulse code modulation channels. Nov. 1996.

Klatt, D. H. (1979). A digital filterbank for spectral matching. In *Proceeding of 1979 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1979)*, 573-576.

Lee, C.-H. (1998). On stochastic feature and model compensation approaches to robust speech recognition. *Speech Communication*, 25, 29-47.

Leggester, C. J. & Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9, 171-186.

Makhoul, J. (1975). Spectral linear prediction: properties and applications. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 23(3), 283-296.

Moreno, P. J., Raj, B., & Stern, R. M. (1996). A vector Taylor series approach for environment-independent speech recognition. In *Proceeding of 1996 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1996)*, 733-736.

Moreno, P. J., Raj, B., & Stern, R. M. (1998). Data-driven environmental compensation for speech recognition: A unified approach. *Speech Communication*, 24(4), 267-285.

Nadas, A., Nahamoo, D., & Picheny, M. (1988). Speech recognition using noise-adaptive prototypes. In *Proceeding of 1988 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1988)*, 517-520.

Sankar, A. & Lee, C.-H. (1996). A maximum-likelihood approach to stochastic matching for robust speech recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 4(3), 190-202.

Segura, J. C., Benitez, M. C., de la Torre, A., Dupont, S., & Rubio, A. J. (2002). VTS residual noise compensation. In *Proceeding of 2002 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2002)*, 1, I-409-I-412.

Tai, C-F. & Hung, J-W. (2006). Silence energy normalization for robust speech recognition in additive noise environments. In *Proceeding of 2006 International Conference on Spoken Language Processing (Interspeech 2006 ─ICSLP)*, 2558-2561.

Tibrewala, S. & Hermansky, H. (1997). Multiband and adaptation approaches to robust speech recognition. In *proceeding of 1997 Eurospeech Conference on Speech Communications and Technology (Eurospeech 1997)*, 2619-2622.

Varga, A. P. & Moore, R. K. (1990). Hidden Markov model decomposition of speech and noise. In *Proceeding of 1990 International Conference on Acoustics, Speech and Signal Processing (ICASSP 1990)*, 845-848.

Yoshizawa, S., Hayasaka, N., Wada, N., & Miyanaga, Y. (2004). Cepstral gain normalization for noise robust speech recognition. In *Proceeding of 2004 International Conference on Acoustics, Speech and Signal Processing (ICASSP 2004)*, I-209-212.