Cluster-based Polynomial-Fit Histogram Equalization (CPHEQ) for Robust Speech Recognition

Shih-Hsiang Lin, Yao-Ming Yeh, Berlin Chen

Department of Computer Science & Information Engineering,
National Taiwan Normal University, Taipei, Taiwan
{69308027, ymyeh, berlin}@ntnu.edu.tw

Abstract

Noise robustness is one of the primary challenges facing most automatic speech recognition (ASR) systems. A vast amount of research efforts on preventing the degradation of ASR performance under various noisy environments have been made during the past several years. In this paper, we consider the use of histogram equalization (HEQ) for robust ASR. In contrast to conventional methods, a novel data fitting method based on polynomial regression was presented to efficiently approximate the inverse of the cumulative density functions of speech feature vectors for HEQ. Moreover, a more elaborate attempt of using such polynomial regression models to directly characterizing the relationship between the speech feature vectors and their corresponding probability distributions, under various noise conditions, was proposed as well. All experiments were carried out on the Aurora-2 database and task. The performance of the presented methods were extensively tested and verified by comparison with the other methods. Experimental results shown that for clean-condition training, our method achieved a considerable word error rate reduction over the baseline system, and also significantly outperformed the other methods.

Index Terms: noise robustness, speech recognition, histogram equalization, polynomial regression model

1. Introduction

Over the last few decades, steady advances have been made in the field of automatic speech recognition (ASR). Most of the current state-of-the-art ASR systems can achieve quite high recognition performance levels in controlled laboratory environments. However, as the systems are moved out of the laboratory environments and put into the real-world applications, noise robustness is actually the primary challenge facing most ASR systems. Robustness techniques in general fall into two main categories [1]: model-space compensation or feature-space compensation, based on where the compensation for the mismatch eventually takes place. Model-space compensation often yields the best recognition performance. It attempts to adjust the acoustic models to accommodate the mismatch caused by noisy environments. Representative techniques, include, but not limited to, maximum a posteriori (MAP) adaptation, maximum likelihood linear regression (MLLR) and parallel model combination (PMC), etc. However, such techniques typically require a sufficient amount of extra adaptation data (either with or without reference transcripts) or a significant computational cost in comparison with feature-space compensation. On the other hand, feature-space compensation is believed to be a simpler and more efficient way to compensate the mismatch caused by noises, and it has also been demonstrated the capability to prevent the degradation of ASR performance under various noisy environments. Well-known techniques include, but not limited to, spectral subtraction (SS), cepstral mean subtraction (CMS), stereo-based piecewise linear compensation for environments (SPLICE), etc. Yet, there has been a tremendous wave of interest in the hybrid techniques of these two categories over the years, such as stochastic vector mapping (SVM), maximum mutual information-SPLICE (MMI-SPLICE), multi-environment model-based linear normalization (MEMLIN), etc.

Among these techniques, CMS is a simple but effective technique for removing the time-invariant distortion introduced by the transmission channel; while a nature extension of CMS, named cepstral mean and variance normalization (CMVN), attempts to normalize not only the means of speech features but also their variances. Although these two techniques have already been proven their effectiveness in compensating the channel distortions and some side effects resulting from additive noises, their linear properties still make them inadequate in tackling the nonlinear distortions caused by various noisy environments [2]. As a remedy to the inherent limitations of CMS and CMVN, histogram equalization (HEQ) has been widely investigated in the recent past [2-7]. HEQ seeks for a transformation mechanism that can map the distributions of the test speech onto predefined (or reference) distributions by utilizing the relationship between the cumulative distribution functions (CDFs) of the test speech and those of the training (or reference) speech [3]. Therefore, HEQ not only attempts to match the means and variances of speech features but also completely match the distributions of speech features between training and test. One additional advantage of HEQ is that it can be easily incorporated with most feature representations and other robustness techniques without the need of any prior knowledge about the actual distortions caused by various kinds of noises.

Recently, we have proposed an efficient method exploring the use of data fitting to approximate the inverse of the CDFs of the speech feature vectors for HEQ, named polynomial-fit histogram equalization (PHEQ), which has also yielded promising results on the Aurora-2 database preliminarily [7]. In this paper, we further extended the use of a single polynomial transformation function in PHEQ into the use of multiple polynomial transformation functions, for each dimension of the feature vectors. The performance of the proposed method was extensively tested and verified by comparison with the other methods.

The remainder of this paper is organized as follows. Section 2 gives a brief review of HEQ. Section 3 elucidates our proposed method in more detail. The experimental settings as well as the evaluation results are presented in Section 4. Finally, conclusions and future work are drawn in Section 5.

2. A Review of Histogram Equalization (HEQ)

2.1. Theoretical Foundation of HEQ

Theoretically, HEQ has its roots in the assumptions that the transformed speech feature distributions of the test (or noisy) data should be identical to that of the training (or reference) data, and each feature vector dimension can be normalized independently of each other. Under the above two assumptions, the aim of HEQ is to find a transformation that can convert the distribution of each feature vector component of the input (or test) speech into a predefined target distribution which corresponds to that of the training (or reference) speech. Accordingly, HEQ attempts not only to match the means and variances of the speech features, but also to completely match the speech feature distributions of training and test data. Put another way, HEQ normalizes all the moments of the probability distributions of the speech features. The equalization can be conducted either in a non-parametric way, such as the table-lookup-based histogram equalization (THEQ) [3-5], which uses a cumulative histogram to approximate the CDF of each utterance [3-5], and the quantile-based histogram equalization (QHEQ) [6], which uses a piecewise transformation function whose parameters are estimated online in a quantile-corrective manner.

2.2. Polynomial-Fit Histogram Equalization (PHEQ)

PHEQ makes use of data fitting (or so-called least squares error regression) to estimate the inverse functions of the CDFs of the training speech. For each speech feature vector dimension of the claen training utterances, given the pair of the CDF value $C_{Train}(x_i)$ of the vector component x_i and x_i itself, the linear polynomial function $G(C_{Train}(x_i))$ with output $\tilde{x}_i$ can be expressed as [7]:

$$G(C_{Train}(x_i)) = \tilde{x}_i = \sum_{m=0}^{M} a_m (C_{Train}(x_i))^m, \quad (1)$$

where the coefficients a_m can be estimated by minimizing the squares error expressed in the following equation:

$$E^2 = \sum_{i=1}^{N} \left(x_i - \sum_{m=0}^{M} a_m (C_{Train}(x_i))^m \right)^2, \quad (2)$$

where N is the total number of training speech feature vectors. During the training phase, the polynomial functions of all dimensions are obtained by minimizing the squares error expressed in Eq. (2). During the recognition phase, for each feature vector dimension, the feature vector components of the test utterance are simply sorted in ascending order of their values to obtain the approximate CDF values, which can be then taken as the inputs to the inverse function to obtain the corresponding restored component values.

The reason why we choose the polynomial function here as the inverse function is mainly because that it has a simple form, without the need of a complicated computation procedure, and has moderate flexibility in controlling the shape of the function. Though the polynomial function is efficient to delineate the transformation function, it is worth mentioning that the polynomial function to some extent has its inherent limitations. For example, high order polynomial functions might lead to over-fitting of the training data. Moreover, the polynomial function would provide good fits for the input data points that are located within the range of values of the training data, but would also probably have rapid deteriorations when the input data points are located outside the range of values of the training data when the order becomes much higher.

3. Cluster-based Polynomial-Fit Histogram Equalization (CPHEQ)

Instead of merely using a single inverse CDF function for the normalization of each feature component, in this paper we investigated the use of multiple inverse CDF functions to obtain the restored value, namely the cluster-based polynomial-fit histogram equalization (CPHEQ). For CPHEQ, not only the clean training utterances but also their corresponding contaminated counterparts that had been corrupted with different kinds of noise types and SNR conditions were used. The contaminated training utterances were first used to train a Gaussian Mixture Model (GMM) whose parameters were estimated by the K-means algorithm followed by the Expectation Maximization (EM) algorithm. Then, each contaminated training speech vector component y_i was assigned to a specific cluster (or a mixture component) k using the following equation:

$$\delta(k \mid y_i) = \begin{cases} 1 & \text{if } k = \arg\max_{k'} p(k' \mid y_i) \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

where $p(k' \mid y_i)$ is the posterior probability of a specific cluster k' given y_i is observed. Furthermore, for each cluster k, a polynomial function $G_k(\bullet)$, defined in analogy with the function $G(\bullet)$ illustrated in Eq. (1), was estimated to restore the contaminated training speech vector components y_i assigned to k to their clean counterparts x_i. The polynomial function $G_k(\bullet)$ with a set of coefficients a_{km} was obtained by minimizing the following squares error E_k^2:

$$E_k^2 = \sum_{i=1}^{N} \left(\left(x_i - \sum_{m=0}^{M} a_{km} (C_{Train}(y_i))^m \right) \times \delta(k \mid y_i) \right)^2. \quad (4)$$

During the recognition phase, each test speech vector component y_j was first assigned to a specific cluster k using Eq. (3) and then was replaced by a restored value $\tilde{y}_j$ using the following equation:

$$\tilde{y}_j = \sum_{m=0}^{M} a_{km} (C_{Test}(y_j))^m. \quad (5)$$

As an alternative approach, the polynomial functions can be derived by minimizing the following square error E_k^2:

$$\hat{E}_k^2 = \sum_{i=1}^{N} \left(\left(x_i - \sum_{m=0}^{M} a_{km} (C_{Train}(y_i))^m \right) \times p(k \mid y_i) \right)^2, \quad (6)$$

where $p(k \mid y_i)$ is the posterior probability of a specific cluster k given y_i is observed. The main different between Eq. (4) and (6) is the amount of data pairs being considered for obtaining the polynomial functions. The estimation using Eq. (4) can be viewed as a hard cluster-assignment approach, where each frame is exactly associated with one cluster. On the other hand, a soft cluster-assignment approach is used in the estimation using Eq. (6), where the error contributed by each data pair is weighted by the corresponding posterior probability of the cluster it probably belongs to. During the recognition phase, the restored value $\tilde{y}_j$ of each test speech vector component y_j therefore can be expressed by:

$$\tilde{y}_j = \sum_{k=1}^{K}\left[\sum_{m=0}^{M} a_{km}\left(C_{Test}\left(y_j\right)\right)^m\right] p\left(k\middle|y_j\right). \quad (7)$$

In recent years, similar efforts also have been made to explore the relationship between the clean training utterances and their contaminated counterparts for speech feature restoration. SPLICE [8] is often considered as a representative of this category. However, the differences between SPLICE and CPHEQ can be discussed from different perspectives. First, SPLICE simply uses a set of linear additive biases (or correction vectors) to approximate the nonlinear relationship between the clean and noisy speech feature vectors, while CPHEQ use a set of polynomial functions to obtain the restored values. Second, SPLICE is only conducted on the speech feature vectors, while not only the speech feature vectors but also their corresponding distribution characteristics are utilized by CPHEQ. Therefore, CPHEQ is believed to be more sophisticated than SPLICE.

4. Experimental Setup and Results

4.1. Experimental Setup

The speech recognition experiments were conducted under various noise conditions using the Aurora-2 database and task [9]. The Aurora-2 database is a subset of the TI-DIGITS, which contains a set of connected digit utterances spoken in English; while the task consists of the recognition of the connected digit utterances interfered with various noise sources at different signal-to-noise ratios (SNRs), in which the Test Sets A and B are artificially contaminated with eight different types of real world noises (e.g., the subway noise, street noise, etc.) in a wide range of SNRs (-5dB, 0dB, 5dB, 10dB, 15dB, 20dB and Clean) and the Test Set C additionally includes the channel distortion. For the baseline system, the training and recognition tests used the HTK recognition toolkit [10].

More specifically, the acoustic model for each digit was a left-to-right continuous density HMM with 16 states, and each state has a six-mixture diagonal GMM. Two additional silence models were defined. One had three states with a six-mixture GMM per state for modeling the silence at the beginning and at the end of each utterance. The other one had one state with a six-mixture GMM for modeling the interword short pause. A 39-dimensional feature vector was extracted at each frame, including 12 Mel-Frequency Cepstral Coefficients (MFCCs) and the logarithm of the energy, as well as their corresponding delta and acceleration coefficients. The training and recognition tests used the HTK recognition toolkit, which followed the setup originally defined for the ETSI evaluations. All the experimental results reported below are based on clean-condition training, i.e., the acoustic models were trained only with the clean (uncontaminated) training utterances.

4.2. Experimental Results

The average word error rate (WER) result obtained by the MFCC-based baseline system is 41.04%, which is an average of the WER results of the test utterances respectively contaminated with eight types of noises under different SNR levels (0db to 20dB) for the three test sets (Sets A, B and C). In the first set of experiments, we evaluated the performance of CPHEQ, for which the polynomial transformation functions were estimated by minimizing the squares error defined in Eq. (4). Different numbers of cluster and different orders of the polynomial functions were extensively investigated, and the associated results are illustrated in Figure 1. It can be found that CPHEQ provides significant performance boosts over the MFCC-baseline system, especially when the number of clusters becomes much larger. However, in the case of large number (e.g., 512 or 1024) of clusters, the performance seems to degrade substantially if the order of the polynomial functions becomes too larger. These results may be explained by the facts that the limited training data used in this study (the fact of the curse of dimensionality), and the use of higher order polynomial functions might lead to oscillations between the exact-fit values. The first row of Table 1 shows the best results summarized from Figure 1 with respect to different numbers of clusters.

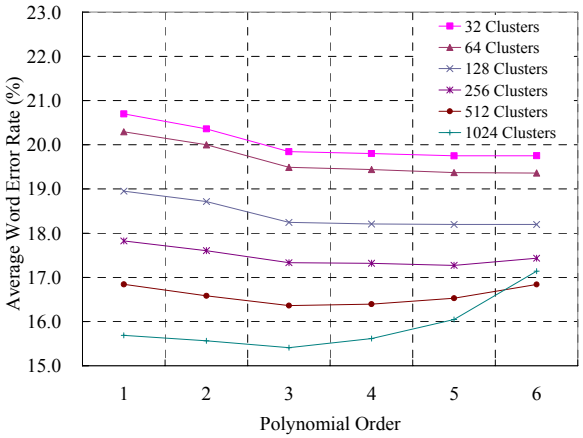

Figure 1: *Average WER results (%) of CPHEQ with respect to different numbers of the clusters and different orders of the polynomial transformation functions.*

	Number of Clusters					
	32	64	128	256	512	1024
Hard	19.73	19.35	18.19	17.27	16.36	15.41
Soft	19.77	19.34	18.19	17.24	16.33	15.40

Table 1. *Comparison of the average WER results (%) between hard cluster-assignment and soft cluster-assignment approaches used for deriving the polynomial transformation functions.*

In the next set of experiments, we investigated the performance of CPHEQ when a different criterion, i.e., Eq. (6), was used to obtain the polynomial transformation functions. The corresponding average WER results are shown in the second row of Table 1. Apparently, when comparing the results of Rows 1 and 2 of Table 1, there is no significant different between them. This might be due to that when the transformation functions are estimated using Eq. (6), the error contributions are prone to be dominated by the cluster with the highest posterior probability for each training speech vector component, which would make the estimation of the transformation functions using Eq. (6) have the same effect as that using Eg. (4). Accordingly, this may also suggest that using Eq. (4) to derive the polynomial functions for CPHEQ is enough and it can also simplify the computation of CPHEQ. As we further compare the best result obtained from Table 1 with the result of the MFCC-based baseline system, it can be found that CPHEQ can provide a relative WER reduction of about 62% over the baseline system.

In the third set of experiments, we compare our proposed approach with the conventional approaches. Table 2 shows the detailed average WER results for test sets A, B and C,

Method	Test A	Test B	Test C	Average
MFCC	41.06	41.52	40.03	41.04
CMVN	27.73	24.60	27.17	26.37
THEQ	20.26	19.45	21.25	20.13
QHEQ	23.74	21.73	23.11	22.81
PHEQ	20.98	20.17	21.43	20.75
SPLICE	17.03	17.12	26.90	19.04
CPHEQ	14.35	14.04	20.28	15.41

Table 2. *Detailed comparison of the WER results (%) obtained by the various approaches.*

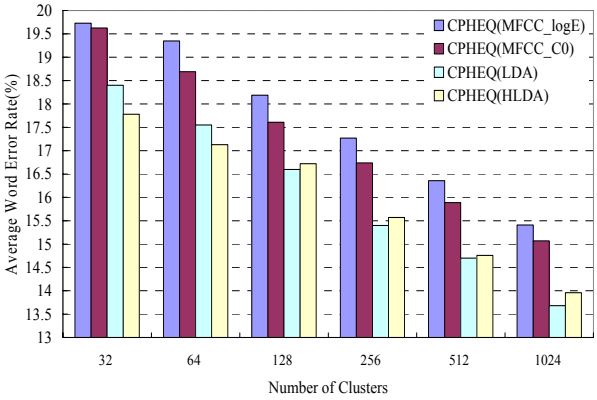

Figure 2: *The average WER results (%) obtained by combing CPHEQ with various feature representations.*

obtained by the MFCC-based baseline system, CMVN, THEQ [3], QHEQ [6], PHEQ [7], SPLICE [8] and CPHEQ, respectively. The number of clusters used in SPLICE was set to 1024 which is the same as that used in CPHEQ. It can be observed that CPHEQ is considerably better than all the other conventional approaches. However, as can also be seen from the last row of Table 2, CPHEQ do not outperform the other approaches significantly on Test Set C. This is mainly due to the fact that Test Set C additionally includes the convolutional distortions which are not seen when training the GMM model of CPHEQ. This will lead to a considerable discrepancy in calculating the posterior probability used in Eq. (3) for the training and test speech. To avoid such a discrepancy, a straightforward remedy is to using CMS to remove the channel distortions. Notice here that CMS was only applied in the training of the GMM model and in the calculation of the posterior probability. This procedure can effectively reduce the WER result of CPHEQ on Test Set C from 20.28% to 16.78%.

Finally, we attempted to combine CPHEQ with four kinds of different feature representations to verify the effectiveness of CPHEQ. The first one is MFCC_logE which followed the original setup defined in the ESTI AURORA evaluations, as described in Section 4.1. The second one is MFCC_C0 which was a modified version of MFCC_logE where the logarithm operation performed on Mel-frequency filter bank outputs was replaced by the 10th root compression, and the 0-th cepstral coefficient was used instead of the logarithmic energy. The third and fourth ones used linear discriminant analysis (LDA) and heteroscedastic linear discriminant analysis (HLDA), respectively. They were both derived directly based on the Mel-frequency filter bank outputs and postprocessed by maximum likelihood linear transform (MLLT) for feature de-correlation. The states of each HMM were taken as the unit for class assignment. The feature vectors from every nine successive frames were spliced together to form the supervectors for the construction of the transformation matrix. The dimension of the resultant vectors was set to 39 [11]. The average WER results are presented graphically in Figure 2, where the results of CPHEQ(MFCC_logE) are actually the results of CPHEQ shown in the first row of Table 1. As Figure 2 indicates, additional performance improvements (absolute word error rate reductions of about 2%) could be obtained when combining with the discriminative feature representations (either LDA or HLDA). The best result was obtained by combing CPHEQ with LDA features, i.e., CPHEQ(LDA). It achieved a relative improvement of about 67% over the MFCC-based baseline system.

5. Conclusions and Future Work

In this paper, we have proposed a cluster-based polynomial-fit histogram equalization (CPHEQ) method for speech feature compensation. The performance of CPHEQ has been extensively tested and verified by comparison with other conventional approaches. Very encouraging results on the Aurora-2 database have been obtained. In the meantime, we are conducting extensive experiments on the use of the mono data (either using the clean or noisy data) instead of the stereo data to derive the polynomial functions for CPHEQ.

6. References

[1] Huang, X. et al., "Spoken Language Processing: A Guide to Theory, Algorithm and System Development," Prentice Hall, 2001.

[2] Torre, A. et al., "Non-Linear Transformations of the Feature Space for Robust Speech Recognition," *in Proc. ICASSP 2002*.

[3] Dharanipargda, S., Padmanabhan, M., "A Nonlinear Unsupervised Adaptation Technique for Speech Recognition," in *Proc. ICSLP 2000*.

[4] Torre, A. et al., "Histogram Equalization of Speech Representation for Robust Speech Recognition," *IEEE Trans. on Speech and Audio Processing* 13, 2005.

[5] Molau, S. et al., "Matching Training and Test Data Distributions for Robust Speech Recognition," *Speech Communication* 41(4), 2003.

[6] Hilger, F. and Ney, H., "Quantile Based Histogram Equalization for Noise Robust Large Vocabulary Speech Recognition," *IEEE Trans. on Audio, Speech and Language Processing* 14, 2006.

[7] Lin, S.-H. et al., "Exploiting Polynomial-Fit Histogram Equalization and Temporal Average for Robust Speech Recognition," *in Proc. ICSLP 2006*.

[8] Deng, L. et al., "Large Vocabulary Speech Recognition under Adverse Acoustic Environments," *in Proc. ICSLP 2000*.

[9] Hirsch, H. G. and Pearce, D., "The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions," *in Proc. ICSLP 2000*.

[10] Young, S. et al., "The HTK Book (for HTK Version 3.3)," Cambridge University Engineering Department, Cambridge, UK, 2005.

[11] Chiu, H. -S., Chen, B., "Word Topical Mixture Models for Dynamic Language Model Adaptation," *in Proc. ICASSP 2007*.