An Effective Feature Compensation Scheme Tightly Matched with Speech Recognizer Employing SVM-based GMM Generation

Wooil Kim, Jun-Won Suh, and John H. L. Hansen

Center for Robust Speech Systems (CRSS), Erik Jonsson School of Engineering & Computer Science
University of Texas at Dallas, Richardson, Texas, USA

{wikim,jxs064200,John.Hansen}@utdallas.edu

Abstract

This paper proposes an effective feature compensation scheme to address a real-life situation where clean speech database is not available for Gaussian Mixture Model (GMM) training for a model-based feature compensation method. The proposed scheme employs a Support Vector Machine (SVM)-based model selection method to effectively generate the GMM for our feature compensation method directly from the Hidden Markov Model (HMM) of the speech recognizer. We also present a strategy to address the case of a combination with Cepstral Mean Normalization (CMN), where the HMM for speech recognizer is obtained using CMN-processed speech database. Experimental results demonstrate that the proposed method is effective at providing a comparable speech recognition performance to the matched data condition where the clean speech database is available for GMM training which is also used for HMM training for speech recognizer. This proves that the SVM-based model selection method is able to effectively generate Gaussian components from the pre-trained HMM model parameters to make the GMM for the feature compensation method be tightly matched to the speech recognizer, resulting in robust speech recognition performance with various types of background noise conditions.

Index Terms: feature compensation, PCGMM, GMM generation, SVM-based model selection, robust speech recognition.

1. Introduction

Presence of background noise generates acoustic mismatch between training and operating conditions of an actual speech recognition system, which is one of the primary factors severely degrading recognition performance. To minimize this mismatch, extensive research has been conducted in recent decades, which includes many types of speech/feature enhancement methods such as Spectral Subtraction, Cepstral Mean Normalization (CMN), and variety of feature compensation schemes. Various model adaptation techniques have been successfully employed such as the Maximum A Posteriori (MAP), Maximum Likelihood Linear Regression (MLLR), and Parallel Model Combination (PMC). Recently, missing-feature methods have shown promising results.

This paper focuses on a feature compensation method employing acoustic model for speech feature space which is usually estimated as a Gaussian Mixture Model (GMM). Multivariate Gaussian-Based Cepstral Normalization (RATZ), Vector Taylor Series (VTS) [1], and Stereo-based Piecewise Linear Compensation for Environments (SPLICE) [2] algorithms can be classified into this GMM-based feature compensation method, and our previous work presented in [3] is also based on GMM for acoustic model. In general, the GMM for the feature compensation methods is obtained by training over clean speech database which is also used for acoustic model training for the speech recognizer (i.e., Hidden Markov Model (HMM)). Acoustically matched models for feature compensation and speech recognition are able to provide the best speech recognition performance.

However, in some real-life situations, training data for the GMM would not be available or training procedure cannot be conducted due to restricted resources. In particular, if a pre-trained HMM for speech recognizer is provided without the clean speech database which was used for the HMM, it is a crucial issue how to obtain the GMM for the feature compensation method which is matched to the HMM in order to get the best recognition performance. In this paper, we will propose an effective strategy to address such a real-life situation for robust speech recognition with background noise presence. Here, we employ our Parallel Combined Gaussian Mixture Model (PCGMM)-based feature compensation method [3], and propose to generate the GMM directly from the HMM of speech recognizer. For effective GMM generation, we employ a Support Vector Machine (SVM)-based model selection method. We will also consider the case of a combination of the feature compensation with CMN for better recognition performance in a real-life situation. The performance will be evaluated using TIMIT database over various types of background noise cases.

This paper is organized as follows. We first review the PCGMM-based feature compensation method as a framework for this study in Sec. 2. Sec. 3 presents the proposed SVM-based model selection method for GMM generation. As a more practical scenario, we will consider the case of CMN combination in Sec. 4. Representative experimental procedures and their results are presented and discussed in Sec. 5. Finally, in Sec. 6 we state the main conclusions of our work.

2. PCGMM-Based Feature Compensation

In the PCGMM-based method [3], the parameters of the noise-corrupted speech model are obtained through a model combination procedure using clean speech and noise models independently [4]. A constant bias transformation of the mean parameters of the clean speech model is assumed in the cepstral domain under the additive noisy environment, which is the assumption generally taken by other data-driven methods [1] as follows,

\[
\mu_{y,k} = \mu_{x,k} + r_k. \tag{1}
\]
where \( \mu_{y,k} \) and \( \mu_{x,k} \) denote mean vectors of \( k \)-th component of GMMs for noise corrupted speech \( y \) and clean speech \( x \) respectively. The bias term \( r_k \) is used for reconstruction of the input speech. The bias term is estimated with Eq.(1), once the mean parameters of the clean speech model and corresponding noise-corrupted speech model are obtained. The MMSE equation for reconstruction of the clean speech is approximated using Eq.(2) as follows [1][3],

\[
\hat{x}_{MMSE} = \frac{1}{N} \int x p(x|y) dx \equiv y - \sum_{k=1}^{K} r_k p(k|y).
\] (2)

The posterior probability \( p(k|y) \) can be calculated based on the parameters of the noisy speech GMM \( \{\omega_k, \mu_{y,k}, \sigma_{y,k}\} \) which consists of \( K \) number of Gaussian components. Fig. 1 presents the block diagram of the PCGMM-based feature compensation as described here.

### 3. GMM Generation Employing SVM-based Model Selection

The large components of Gaussians can be summarized by most distinctive Gaussians. The Gaussian selection requires good distance measure between Gaussians. In this study, the SVM ranking method is employed to measure the distance between target Gaussian and available Gaussian candidates. The data-driven background dataset selection has been developed showing improvement in a speaker recognition task [5]. In our model selection scheme, this method is used for measuring the distance of Gaussians.

The SVM training requires target and non-target classes, and the positive and negative examples representing each class. The positive example represents the target Gaussian, and the last of Gaussians become a negative example. The SVM kernel enables each example to project in higher dimension so that the resulting support vectors maximize the distance between target and non-target classes. The Gaussian components (mean, variance, and weight) are concatenated to build the vector, and this vector is used as one example for both classes. The SVM trains one target class with one positive example and negative examples, the resulting target support vector for one target class can measure the closeness by scoring same negative examples used for training stage. More close to target class within negative examples produces higher score than placed further away from target class. The distance measure is performed for each target class/Gaussian, and then the negative examples are rank by descending order on scores. After doing distance measure for all Gaussians, the most occurred Gaussians are selected until user specified numbers.

This study focuses on selecting most informative Gaussians from a pool of Gaussian components which are collected from a pre-trained HMM for speech recognizer. Each Gaussian of the pool represents one example. We investigated the performance of the GMM generation by changing the combination of parameters (mean vector, variance, and weight) for an example. The Jackknife approach is used for making negative examples for each positive example [6]. Fig. 2 illustrates a flow diagram of the proposed speech recognition scheme with the PCGMM-based method employing the GMM generation method.

### 4. Combination with Cepstral Mean Normalization

Speech recognition performance is generally considered to improve when combined with CMN processing. Therefore in many cases, the HMM for speech recognizer is obtained from the training database which is processed with the CMN. When the CMN-processed GMM is only available from the CMN-processed HMM, how to apply the obtained GMM to the PCGMM method will be a crucial issue, since the model parameters for the CMN-processed GMM are not mathematically suitable to the model combination procedure for the PCGMM-based method.

As an approximation method to address the CMN-processed GMM, first we can consider to employ a global mean parameter. From an available clean speech database, we can obtain a global mean in the cepstral domain. By adding the global mean to the all mean vectors of the CMN-processed GMM (in the cepstral domain), the model combination procedure can be conducted for the PCGMM-based feature compensation method. However, this method requires an additional clean database for obtaining the global mean, which is not suitable to our study where no additional speech database is assumed. As another approximation method, we can employ an utterance mean parameter of every input speech instead of the global mean. However, the utterance mean of input speech will include noise corruption in case of background noise presence. In our study, we apply the spectral subtraction [7] to the input speech and then obtain an utterance mean to minimize the noise interference for estimating a clean speech mean vector. The performance evaluation of the proposed approximation methods for combination of PCGMM and CMN methods will be discussed in the followed section.

### 5. Experimental Results

#### 5.1. Experimental Setup and Baseline Performance

The TIMIT speech corpus was used for performance evaluation of the proposed method. A total of 4.1 hours of speech (462 speakers, 4,620 utterances) were used for training, and 1.5 hours of data (168 speakers, 1,680 utterances) were used for...
The training and the test sets do not overlap each other in speakers and uttered sentences. The data was down-sampled to 8kHz, so that each speech sample contains 4kHz full-band frequency. In order to evaluate the performance under various types of background noise conditions, noise corrupted test sets were generated by combining clean speech samples with car noise, factory noise, speech babble, and background music audio samples. The car noise, factory noise and speech babble samples were obtained from NOISEX92, and the background music samples consist of prelude parts of ten Korean popular songs with varying degrees of beat and tempo. Each test set consists of 1,680 utterances at three different SNRs: 5, 10, and 15 dB.

We employed SPHINX3 [8] as the HMM-based speech recognizer to obtain recognition accuracy in background noise conditions. Each HMM represents a tri-phone which consists of 3 states with an 8-component GMM per state, which is tied with 1138 states. The task has 6233 words as the vocabulary, and the trigram language model is adapted on the TIMIT database using a Broadcast News language model as an initial model. A conventional MFCC (Mel-Frequency Cepstral Coefficient) feature front-end is employed in the experiment, which was suggested by the European Telecommunication Standards Institute (ETSI) [9]. An analysis window of 25msec in duration is used with a 10msec skip rate for 8kHz speech data. The computed 23 Mel-filterbank outputs are transformed to 13 cepstrum coefficients including c0 (i.e., c0-c12).

Performance of the baseline system (no compensation) was examined with comparison to several existing pre-processing algorithms in terms of speech recognition performance. Spectral Subtraction (SS) [7] combined with Cepstral Mean Normalization (CMN) was selected as one of the conventional algorithms. They represent some of the most commonly used techniques for additive noise suppression and removal of channel distortion respectively. We also evaluated the Vector Taylor Series (VTS) algorithm for performance comparison [1]. The Advanced Front-End (AFE) algorithm developed by ETSI was also evaluated as one of the state-of-the-art methods, which contains an iterative Wiener filter and blind equalization [10]. Table 1 demonstrates speech recognition performance (i.e., Word Error Rate, WER) of the baseline system and the conventional algorithms on all background noise conditions. Here, we obtained 50.91%, 36.70%, 37.31%, and 29.55% for baseline (no processing), SS+CMN, VTS, and AFE as average WERs over 5, 10, and 15 dB SNRs of all four noise conditions.

5.2. Performance Evaluation of the GMM Generation Methods

Table 2 shows the performance of speech recognition with the PCGMM-based feature compensation method as a change of training database used for the GMM of the PCGMM method. The estimated GMM consists of 256 Gaussian components. Here we obtained 28.87% in average WER over all noise types and SNRs when employing TIMIT database for GMM training which is also used for training the HMM model for speech recognizer. This case is an ideal situation for feature compensation method where it is assumed that the clean speech database is available for GMM training for feature compensation method which is matched with HMM training for speech recognizer. For mismatched condition for GMM training, we used AURORA 2.0 [11] which consists of utterances of English digits. In case of GMM training using the AURORA database, 32.49% average WER was obtained, which degrades by 3.62% compared to the matched condition (i.e., GMM training over TIMIT database). The performance degradation can be explained as mismatched acoustic feature space between GMM for feature compensation and HMM for speech recognition, which were obtained by training over AURORA and TIMIT corpus respectively (i.e., digits vs. 6233-word vocabulary).

Table 3 shows the performance evaluation employing the GMM generation methods which are proposed in this paper. The GMM for the PCGMM-based feature compensation is generated by different types of GMM generation methods directly from the HMM model for speech recognizer. Therefore it is expected that the obtained GMM will be well matched with the HMM for speech recognition in acoustic feature space. First, we collected the Gaussian components from the HMM of the speech recognizer. As mentioned earlier, the triphone model is tied with 1138 number of states and each state consists of a GMM with 8 Gaussian components. For the “Euclidean” method in Table 3, we merged all 9104 (= 8 × 1138) Gaussians into 256 components, using Euclidean distance measure and the binary splitting algorithm which commonly used for vector quantization and Gaussian clustering methods. Using the GMM model obtained by the Euclidean-based method, 30.36% average WER was obtained, which is better than the mismatch case (i.e., AURORA database in Table 2), however, it is still worse than the case of matched training data (i.e., TIMIT database in Table 2).

As a similar manner with Euclidean approach, 256 Gaussian components are selected by the proposed SVM-based scheme from the collected pool of the Gaussian components. For “SVM1”, the example for the SVM selection is made by concatenation of the mean vector, variance vector, and weight of each Gaussian. The example for “SVM2” consists of mean and weight of each Gaussian. By employing the proposed SVM based methods (SVM1 and SVM2), we obtained 28.46% and 28.23% average WERs which both are slightly better than the matched case. We believe that, the GMM obtained by employing the proposed GMM generation method for model-based feature compensation is well matched with the speech recognizer, leading to comparable (even slightly better) performance to the matched training data condition. It can be considered that the SVM-based method provides better performance compared to the Euclidean method, since the kernel projects the each example into higher dimension instead of summing up differences in each dimension.
Table 3: Recognition performance with the PCGMM-based method employing different types of GMM generation methods as average WERs over all SNRs (%).

<table>
<thead>
<tr>
<th>GMM generation</th>
<th>Car</th>
<th>Factory</th>
<th>Babble</th>
<th>Music</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>38.45</td>
<td>30.28</td>
<td>31.44</td>
<td>21.27</td>
<td>30.36</td>
</tr>
<tr>
<td>SVM1</td>
<td>32.42</td>
<td>26.58</td>
<td>32.49</td>
<td>22.35</td>
<td>28.46</td>
</tr>
<tr>
<td>SVM2</td>
<td>31.69</td>
<td>25.92</td>
<td>32.02</td>
<td>23.27</td>
<td>28.23</td>
</tr>
</tbody>
</table>

Table 4: Recognition performance with the PCGMM-based method in case of CMN combination as average WERs over all SNRs (%).

<table>
<thead>
<tr>
<th>SCGMM-CMN</th>
<th>GMM train</th>
<th>Car</th>
<th>Factory</th>
<th>Babble</th>
<th>Music</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional</td>
<td>Clean TIMIT</td>
<td>31.50</td>
<td>25.75</td>
<td>27.97</td>
<td>19.30</td>
<td>26.11</td>
</tr>
<tr>
<td>GMean</td>
<td>CMN+TIMIT</td>
<td>31.81</td>
<td>26.00</td>
<td>27.98</td>
<td>18.74</td>
<td>26.13</td>
</tr>
<tr>
<td>UMean</td>
<td>CMN+TIMIT</td>
<td>32.79</td>
<td>27.20</td>
<td>29.02</td>
<td>20.10</td>
<td>27.28</td>
</tr>
<tr>
<td>UMean</td>
<td>CMN+TIMIT</td>
<td>36.31</td>
<td>30.03</td>
<td>30.41</td>
<td>21.05</td>
<td>29.45</td>
</tr>
</tbody>
</table>

5.3. Performance Evaluation in case of CMN Combination

Next we will see the PCGMM-based feature compensation method combined with CMN. In this section the HMM of the speech recognizer is obtained by training over CMN-processed speech database. First, as a baseline performance, the input speech (cepstral feature vector) is reconstructed by the CMN-based method, and then processed by the CMN ("Conventional" in Table 4). This requires clean speech data (without CMN) for the GMM training for the PCGMM method. We obtained 26.13% average WER which is better than the PCGMM solely used case in Table 2 (28.87%). The second to fourth rows in Table 4 indicate the cases of the PCGMM method with the CMN-processed GMM. Here the CMN-processed GMMs were obtained by training over the CMN-processed TIMIT or AURORA database. For the model combination procedure in the PCGMM method with CMN-processed GMM, global mean parameter (GMean) or utterance mean (UMean) is used. The global mean parameter is obtained from the training database used for obtaining the GMM, and the utterance mean is obtained from every input speech. Even though there was a somewhat decrease in performance, the use of utterance mean generates a comparable WER to the ideal and the global mean cases (27.28% vs. 26.13%). As a similar trend with the results in Table 2, the mismatched training database case (UMean with AURORA) is worse than the matched case (UMean with TIMIT). The results here demonstrate the use of utterance mean is effective at employing the CMN-processed GMM for the PCGMM method.

We also evaluated the recognition performance employing the proposed SVM-based GMM generation method in case of the CMN combination. For the model combination of the PCGMM method, utterance mean is used for all three experiments in Table 5, resulting in no additional training data for GMM training. From the results, SVM2 method generated 26.53% average WER which is comparable to the baseline matched training data case (26.13% of PCGMM+CMN) in Table 4.

Table 5: Recognition performance with the PCGMM-based method employing different types of GMM generation methods in case of CMN combination as average WERs over all SNRs (%).

<table>
<thead>
<tr>
<th>GMM Generation</th>
<th>Car</th>
<th>Factory</th>
<th>Babble</th>
<th>Music</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean</td>
<td>37.14</td>
<td>30.23</td>
<td>29.24</td>
<td>19.59</td>
<td>29.05</td>
</tr>
<tr>
<td>SVM1</td>
<td>31.04</td>
<td>25.54</td>
<td>29.02</td>
<td>20.68</td>
<td>26.57</td>
</tr>
<tr>
<td>SVM2</td>
<td>31.14</td>
<td>25.85</td>
<td>28.66</td>
<td>20.48</td>
<td>26.53</td>
</tr>
</tbody>
</table>

6. Conclusions

In this study, an effective feature compensation scheme was proposed to address a real-life situation where clean speech database is not available for GMM training for model-based feature compensation. The proposed scheme employed a SVM-based model selection method to effectively generate the GMM for the feature compensation method directly from the HMM employed by the speech recognizer. We also presented a strategy to address the case where the pre-trained HMM was obtained using CMN-processed speech database. Experimental results demonstrated that the proposed method is effective at providing comparable speech recognition performance to the matched training data condition where the clean speech database is available for GMM training which is also used for HMM training speech recognizer. This proved that the SVM-based model selection method is able to effectively generate Gaussian components from the pre-trained HMM model parameters to make the GMM for the feature compensation method be well matched to the speech recognizer, resulting in robust speech recognition performance.

7. References