HMM-BASED FEATURE COMPENSATION METHOD:
AN EVALUATION USING THE AURORA2

Akira Sasou\textsuperscript{1}, Futoshi Asano\textsuperscript{1}, Kazuyo Tanaka\textsuperscript{2,1}, Satoshi Nakamura\textsuperscript{3}

\textsuperscript{1}National Institute of Advanced Industrial Science and Technology (AIST)
a-sasou@aist.go.jp, f.asano@aist.go.jp
\textsuperscript{2}Institute of Library and Information Science, University of Tsukuba
ktanaka@ulis.ac.jp
\textsuperscript{3}ATR Spoken Language Translation Research Laboratories
satoshi.nakamura@atr.jp

ABSTRACT
In this paper, we describe an HMM-based feature-compensation method. The proposed method compensates for noise-corrupted features in the MFCC domain using the output probability density functions (pdf) of the Hidden Markov Models (HMM). In compensating the features, the output pdfs are adaptively weighted according to forward path probabilities. Because of this, the proposed method can minimize degradation of feature-compensation accuracy due to temporally changing noise environment. We evaluated the proposed method based on the AURORA2 database. All the experiments were conducted in a clean condition. The experiment results indicate that the proposed method, combined with cepstral mean subtraction, can achieve a word accuracy of 87.64\%. We also show that the proposed method is useful in a transient pulse noise environment.

1. INTRODUCTION
One approach to realize noise-robust speech recognition is feature compensation of noise-corrupted speech during the front-end processing\cite{1,2}. Segura et al. demonstrated the effectiveness of the feature-compensation method based on Gaussian Mixture Model (GMM), which estimates the expectation of distortion from a noise-corrupted speech feature based on the GMM, and generates a compensated speech feature by subtracting the estimated distortion from the corrupted speech feature\cite{3}. The GMM is prepared in advance by being trained from clean-speech features. In the feature-compensation process, the GMM is adapted to the noisy environment by using several frames prior to an utterance, which are expected to be noise-only frames. Because the adapted GMM is well matched to noise-corrupted speech features, the method can adequately compensate for the corrupted speech features in a stationary noise environment. However, because the noise characteristics change temporally in a non-stationary noise environment, mismatches between the adapted GMM and a current noise-corrupted speech feature occur. Because of this mismatch, the post probabilities of Gaussians, which should not contribute to compensation for features, are overestimated. This causes the degradation of the feature-compensation accuracy.

To avoid this problem, we propose a Hidden Markov Model (HMM) based feature-compensation method. The proposed method assumes that the non-stationary noise can be divided into a stationary component and some temporally changing components. The effects of the stationary component are eliminated by adapting the output pdfs of the HMMs to the stationary component of the noise like the GMM-based methods do. With regard to temporally changing components, the proposed method can eliminate the degradations of feature-compensation accuracy by evaluating the post probability of each Gaussian based on the Gaussians adaptively weighted according to forward path probabilities. Even if the current noise characteristics have changed temporally, the proposed method can enhance the post probabilities of Gaussians, which should contribute to feature compensation.

2. FEATURE COMPENSATION BASED ON HMM
In the following, $S$ denotes the entire set of HMM states, $\pi_s$ denotes the initial probability of the $s$th state, and $a_{sj}$ denotes the transition probability from the $j$th state to the $q$th state. The output probability density function (pdf) of the $j$th state $b_j(x)$ is given by

$$b_j(x) = \sum_{m=1}^{M} w_{jm} \mathcal{N}(x; \mu_{jm}, \Sigma_{jm})$$

(1)

where $M$ is a number of Gaussians and $\Sigma_{jm}$ is a diagonal matrix. The HMM parameters are estimated from a clean training data set.

The first step in the feature-compensation process is to adapt all the output pdfs of the HMM to a stationary component of noise. This pdf-adaptation process is executed for each utterance by using $N$ frames prior to the utterance, which are expected to be noise-only frames. Let $x$ denote an element vector consisting of a part of an observed feature vector $x$ that is to be compensated. The elements of the vector $x$, except the vector $x$, remain the...
same. In our settings, the vector $x$ has a 39-dimensional vector, which consists of a 13-dimensional base Mel-Frequency Cepstral Coefficient (MFCC), its delta and delta-delta components, and the element vector $\hat{x}$ corresponds to the base MFCC. Also, let $\mathcal{N}(\mu_{jm}, \Sigma_{jm})$ represent the $m$th Gaussian corresponding to the element vector $\hat{x}$ in the output pdf of the $j$th state. The distortion component $d_{jm}$ of each Gaussian $\mathcal{N}(\mu_{jm}, \Sigma_{jm})$ is then evaluated according to the following equation.

$$d_{jm} = \frac{1}{N} C \cdot \sum_{t=1}^{N} \log \left[ 1 + \exp \left\{ C^{-1} \cdot (\hat{x}_t - \mu_{jm}) \right\} \right]$$

(2)

where $C$ denotes the Discrete Cosine Transform (DCT) matrix. Every noise-adapted Gaussian $\mathcal{N}(\hat{m}_{jm}, \hat{Z}_{jm})$ is given by

$$\hat{m}_{jm} = \mu_{jm} + \hat{d}_{jm}$$

$$\hat{Z}_{jm} = \Sigma_{jm}.$$  

(3)

The proposed method utilizes each cumulative probability $\alpha(s, t)$ of the best path to the $s$th state at time $t$ for the feature-compensation process. Every cumulative probability $\alpha(s, t)$ is recursively calculated using the Viterbi algorithm as follows. First, the initial probabilities are set to $\alpha(s, 0)$.

$$\alpha(s, 0) = \log(\pi_s) \text{ for } \forall s \in S$$

(4)

For each time instant $t$ and for each state $s$, the cumulative probabilities are then calculated as follows.

$$\alpha(s, t) = \max_{j \in S} \alpha(j, t-1) + \log\{a_{js} \cdot b_j(y_t)\}$$

(5)

where the vector $y_t$ represents the compensated feature vector. At the last time instant $T$, the most likely final state is selected according to the following equation.

$$\log P_{\text{max}} = \max_{s \in S_T} \alpha(s, T)$$

(6)

where $S_T$ represents the set of the final states.

Each compensated-feature vector $y_t$ is evaluated according to the following procedures, which are executed for each time instant $t$ before the cumulative probabilities are updated using Eq.(5). First, by using the previously calculated cumulative probabilities, we evaluate weights $\alpha'(s, t-1)$ to be applied to the noise-adapted output pdfs of all the states.

$$\alpha'(s, t-1) = \exp(\alpha(s, t-1)/t)$$

(7)

The posterior probability $P(j, m)$ of each noise-adapted Gaussian $\mathcal{N}(\hat{m}_{jm}, \hat{Z}_{jm})$ on condition of a given observation $x_t$ is then evaluated using the following equation.

$$P(j, m) = \frac{\alpha'(j, t-1)w_{jm}\mathcal{N}(\hat{x}_t; \hat{m}_{jm}, \hat{Z}_{jm})}{\sum_{s \in S} \sum_{m=1}^{M} \alpha'(s, t-1)w_{sn}\mathcal{N}(\hat{x}_t; \hat{m}_{sn}, \hat{Z}_{sn})}$$

(8)

The compensated element vector $\hat{y}_t$ is given by

$$\hat{y}_t = \hat{x}_t - \sum_{j \in S} \sum_{m=1}^{M} P(j, m) \hat{d}_{jm}.$$  

(9)

When the noise has no stationary component, like noises consisting of only transient pulses, the distortion components of Gaussians are always equal to zero; $d_{jm} = 0$. Hence, the feature compensation of Eq.(9) becomes useless. In this case, we use the following feature compensation instead of Eq.(9).

$$\hat{y}_t = \sum_{j \in S} \sum_{m=1}^{M} P(j, m)\mu_{jm}$$

(10)

The compensated feature vector $y_t$ is finally obtained by combining the compensated element vector $\hat{y}_t$ with the elements of the observed feature vector $x_t$ except the element vector $x_t$.

3. EXPERIMENT RESULTS

3.1. Experiment Setup

We used the AURORA2 database[4] to evaluate the proposed method, and adopted the following procedures for feature extraction instead of using the AURORA front-end (version 2.0). The frame length is set at 25 ms and the period at 10 ms. The FFT is calculated after pre-emphasizing by $1-0.97z^{-1}$. The inner products between the squared amplitudes of the FFT coefficients and the triangle windows of the Mel Filter Bank are calculated to generate the Mel Filter Bank Energy (Mel-FBE) feature. The MFCC is then obtained by applying DCT to the natural logarithm of the Mel-FBE feature. The MFCC is a 13-dimensional vector including the 0th coefficient. The delta and the delta-delta features are evaluated. Combining these features generates the 39-dimensional feature vector. The clean acoustic models for digits (1-9, zero, oh) were composed of 16 emitting states, with twenty mixtures per state. Those of sil were composed of three emitting states, and one emitting state for sp, with thirty-six mixtures per state for both sil and sp. These clean acoustic models were trained using only clean-speech data (clean training condition). Ten frames were used for adapting the output pdfs to the noise environment.

The proposed feature-compensation method can be implemented in two ways. One achieves feature compensation and decoding simultaneously. The other achieves these two processes separately. We call the former Method-1, and latter Method-2, and describe the details below.

- **Method-1**
  Method-1 compensates the features according to the procedures described in section 2. Because the feature-compensation process is embedded into the decoding process, these two processes end simultaneously. Also, the two processes are implemented by using only the HMMs that the decoder has in advance. Therefore, Method-1 needs no extra-parameters for making the decoder robust against effects of additive noises as the GMM-based methods do. However, one drawback of this method is using HMMs that have not been trained from the compensated features for calculating output probabilities of the compensated features.
Table 1: Word accuracy of Method-1 evaluated on the AURORA2 database.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>96.63</td>
<td>96.49</td>
<td>96.61</td>
<td>96.61</td>
</tr>
<tr>
<td>10dB</td>
<td>96.74</td>
<td>96.61</td>
<td>96.66</td>
<td>96.66</td>
</tr>
<tr>
<td>15dB</td>
<td>97.35</td>
<td>97.48</td>
<td>97.68</td>
<td>97.58</td>
</tr>
<tr>
<td>18dB</td>
<td>93.65</td>
<td>93.62</td>
<td>93.68</td>
<td>93.68</td>
</tr>
<tr>
<td>Sub</td>
<td>81.98</td>
<td>81.24</td>
<td>81.68</td>
<td>81.68</td>
</tr>
<tr>
<td>Average</td>
<td>93.43</td>
<td>90.28</td>
<td>90.15</td>
<td>90.15</td>
</tr>
</tbody>
</table>

Table 2: Word accuracy of Method-2 evaluated on the AURORA2 database.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>96.63</td>
<td>96.49</td>
<td>96.61</td>
<td>96.61</td>
</tr>
<tr>
<td>10dB</td>
<td>96.74</td>
<td>96.61</td>
<td>96.66</td>
<td>96.66</td>
</tr>
<tr>
<td>15dB</td>
<td>97.35</td>
<td>97.48</td>
<td>97.68</td>
<td>97.58</td>
</tr>
<tr>
<td>18dB</td>
<td>93.65</td>
<td>93.62</td>
<td>93.68</td>
<td>93.68</td>
</tr>
<tr>
<td>Sub</td>
<td>81.98</td>
<td>81.24</td>
<td>81.68</td>
<td>81.68</td>
</tr>
<tr>
<td>Average</td>
<td>93.43</td>
<td>90.28</td>
<td>90.15</td>
<td>90.15</td>
</tr>
</tbody>
</table>

Method-2
To overcome the drawback of Method-1, Method-2 performs feature compensation and decoding separately. The feature-compensation process is implemented according to the procedures described in section 2. Method-2 differs from Method-1 in that it extracts only the base MFCC compensated by the process, and discards the recognition results. The new feature vectors are then generated by combining the compensated base MFCC with its re-evaluated delta and delta-delta components. In evaluating the output probabilities of the decoding process, Method-2 uses HMMs that were trained from features compensated by the same procedures. Method-2 requires preparing extra HMMs in addition to the HMMs that the decoder already has.

3.2. Experiment results of Methods-1 and -2
Tables 1 and 2 illustrate the word accuracy for Method-1 and Method-2. To demonstrate the effectiveness of the proposed method, we conducted an experiment using our front-end feature extraction without any compensation process. Table 3 is a summary of the baseline word accuracy, which was evaluated by averaging over an SNR range from 0 to 20dB for each test set. Using this baseline performance, we evaluated the relative improvement according to the following equation.

\[ R.I. = \frac{\text{Result} - \text{Baseline}}{\text{Baseline}} \times 100\% \]  \hspace{1cm} (11)

Relative improvements of Method-1 and Method-2 are presented in Table 4. From these results, we can see that Method-1 obtained an overall relative improvement of 51.84%, without any additional parameters for feature compensation. Also, by adopting the corresponding HMM to the compensated features, Method-2 can achieve an overall relative improvement of 59.25%.

3.3. Experiment results of Method-2 followed by CMS
We conducted experiments combining cepstral mean subtraction (CMS) with the proposed methods. The preliminary experiments indicated that CMS should be applied to the features compensated by the proposed method because there were negative effects when CMS is executed prior to the proposed methods. Method-1 cannot easily apply CMS to the compensated features because the compensation and decoding processes are executed simultaneously. We therefore conducted an experiment with Method-2 followed by CMS. One HMM used for feature compensation was trained from the original features without any compensation. The other HMM used for the decoding process was trained from the features compensated by Method-2, followed by CMS. Table 5 and Table 6 illustrate the full absolute results and the relative performance. By combining Method-2 with CMS, this feature-compensation method can achieve an overall relative improvement of 66.24%.
3.4. Comparison of GMM-based method

To compare the proposed method with the GMM-based feature-compensation method, we conducted experiments using GMM followed by CMS as front-end feature compensation. The Number of Gaussians ranged from 64 to 4096. Each back-end HMM was trained from the compensated features of each condition.

Table 7 contains the obtained absolute word accuracy of the GMM-based method. From these results, we can see that the word accuracy of the GMM-based method tends to saturate when the number of mixtures exceeds 512. Alternatively, a total number of 3664 Gaussians was used in the HMMs for Method-2, and Method-2 followed by CMS achieved a word accuracy of 87.64% as denoted in Table 5. The word accuracy of the proposed method is greater than that of GMM4096+CMS. That is, the proposed method can improve the word accuracy more than the GMM-based method in spite of having fewer Gaussians. These results indicate that it is beneficial to adaptively weight distributions according to forward path probabilities.

3.5. Evaluation in a transient pulse noise environment

In the experiment conducted in a transient pulse noise environment, we used wooden collision sound sources recorded in RWCP Sound Scene Database[5]. We generated noise-corrupted speech data by adding a transient pulse every 125ms to clean speech data of AURORA2. Noise-corrupted features were compensated and recognized by Method-1 in which Eq.(10) was adopted for feature compensation. The feature-compensation process was applied to both the base component and to the delta and delta-delta components. We also conducted an experiment for evaluating the baseline performance. The HMMs used in both the experiments are the same. Table 8 shows the results. From these results, we can see that the proposed method is able to compensate for features corrupted by transient noise pulses.

4. CONCLUSION

In this paper, we presented an HMM-based feature-compensation method, which can minimize degradation of feature-compensation accuracy due to temporally changing noise environment by compensating for features based on distributions adaptively weighted according to forward path probabilities. The experiment results reveal that the proposed method can improve the word accuracy more than the GMM-based method in spite of containing fewer Gaussians than the GMM-based method. We also confirmed that the proposed method is useful in a transient pulse noise environment.

5. REFERENCES