NOISE ROBUST DIGIT RECOGNITION WITH MISSING FRAMES

Cenk Demiroglu and David V. Anderson

School of ECE, Georgia Institute of Technology, Atlanta, GA, 30332-0250

ABSTRACT

Noise robustness is one of the most challenging problems in speech recognition research. In this work, we propose a noise robust and computationally simple system for small vocabulary speech recognition. We approach the noise robust digit recognition problem with the missing frames idea. The key point behind the missing frames idea is that frames with energies below a certain threshold are considered unreliable frames. We set these frames to a silence floor and treat them as silence frames. Performing this operation only in decoding stage creates high mismatch between trained speech and decoded speech. To solve the mismatch problem, we apply the same thresholding algorithm on the training data before training. The algorithm adds a negligible computational complexity at the front end, and decreases the overall computational complexity. Moreover, it outperforms other computationally comparable, well known methods. This makes the proposed system particularly suitable for real-time systems.

1. INTRODUCTION

Noise robustness is one of the most challenging problems in speech recognition research. The main problem is the acoustic mismatch between the training and testing environments due to background noise in the testing environment, which typically does not exist in the training environment. This mismatch causes a significant drop in the recognizer performance (measured as word accuracy rate). An obvious approach to attack this problem is to have a separate training set for each noise type. The problem with this approach is that the large number of different background noise types possible in a realistic environment makes it impractical to use this method.

Noise robustness for speech recognition has been investigated by researchers for a long time and significant amount of work has been done [1]. One of the commonly used strategies is to look for robust features that will not be significantly affected in a noisy environment while having good discrimination capability [1]. Another approach is to use a speech enhancement algorithm to suppress the noise before feeding the signal to the recognizer [1]. There are many speech enhancement algorithms in the literature but most of them are designed to increase the perceptual quality of the speech and have not been found to perform well when used with a recognizer.

Besides the algorithms that focus on the front end, there are also algorithms that try to solve the problem by modifying the acoustic models. One of the well known algorithms of this type is Parallel Model Combination (PMC) [1]. PMC combines a statistical model of the noise with the acoustic models of speech to generate models that match the environment. Adaptation–based schemes like Maximum Likelihood Linear Regression (MLLR) constantly adapt the models to the environment as new noisy speech is gathered. These systems perform well relative to other approaches; however, model–based algorithms suffer from high computational complexity.

In this paper we target low–power, real–time systems such as might be used in embedded devices. For such systems, the complexity of PMC and other model–based algorithms makes their deployment infeasible, especially when the background noise is nonstationary.

Missing feature theory is another alternative that has been shown to improve the performance in noisy environments [2]. The idea is to discard the features that are known to be masked by noise. One problem with this approach is the difficulty in choosing which features to discard. Especially in low SNR cases, it is difficult to estimate the SNR at each frequency band, which makes it very difficult to assess the reliability of the band. Another important problem is that the computational complexity doubles with this method.

In this work, we propose a new algorithm to reduce the effects of noise on the performance of the speech recognizer for a small vocabulary task. Instead of trying to estimate the reliable features, we estimate the reliability of the whole speech frame by using an energy threshold. The frames that have energies below the threshold are set to silence level, while the reliable frames are kept as they are. The key idea behind this is the observation that high energy speech frames will not be influenced much by the noise while the low energy and silence periods are dramatically distorted at low SNRs. The additional computational complexity at
the front end is almost negligible. Furthermore, the average time spent for Viterbi search is significantly decreased with the proposed system compared to unprocessed case. Therefore, it is highly attractive for low power, real time systems like portable devices.

Voice activity detection (VAD) is shown to be effective in reducing the insertion errors. The difference of the proposed system is that it not only floors the non-speech segments but also floors the low energy speech segments. This improves the robustness and performance of the system.

This paper is organized as follows. In the next section effects of noise on speech features are discussed for Mel Frequency Cepstrum Coefficients (MFCCs). In section 3, we describe our proposed system. In section 4, experiment results are presented. Finally we make a conclusion and discuss possible future directions in section 5.

2. EFFECT OF NOISE ON MEL FREQUENCY CEPSTRUM COEFFICIENTS

![Block diagram of extracting MFCCs from speech input.](image1)

The energy of the signal at the output of each filterbank is passed through a logarithmic nonlinearity. The logarithmic energies are decorrelated using a Discrete Cosine Transform (DCT). The outputs of the DCT operation are called MFCCs.

Figure 1 shows the block diagram of the feature extraction algorithm used for extracting MFCCs. The speech signal is analyzed with Short Time Fourier Transform (STFT) with a typical window length of 20 msec. The spectrum of the windowed speech is passed through a filterbank.

![Log output of the first filterbank for clean and noisy speech at 0dB.](image2)

Figure 3. Illustration of the thresholding algorithm. The output of the first filterbank is shown for clean speech, noisy speech, and thresholded noisy speech at 0 dB.

The observation done in the previous section leads us to ignore some parts of the data where signal power is low, which corresponds to silence and low energy speech segments. Thus in this work, we looked at noise robustness as a missing data problem rather than a noisy data problem. Some parts of the spectrum can be reliable if it is narrowband noise and can be used for the recognition task [2]. The problem with this approach is the difficulty in estimating the reliability of each band. We approached this problem by trying to assess the reliability of the whole frame rather than the individual bands. This is a relatively easier task and does not make any assumptions about the noise.

The speech frames that are above a certain energy threshold are considered reliable in this work. The frames with energy lower than this threshold are considered unreliable and their filterbank outputs are set to silence level. Figure 3 illustrates an example of this method. The threshold is set to $\exp(14)$ in this figure and the speech is contaminated by babble noise at 0 dB. The distortion at low energy
periods are substantially reduced and the distortions at low power speech periods are almost the same after thresholding.

Choosing a threshold has two main challenges. The first one is balancing the trade off between average distortion and data loss. Increasing the threshold increases the reliability of the features when the energy is above the threshold, while it also increases the number of frames that are assumed unreliable and therefore set to silence level. It is crucial to choose a good operating point at each SNR for the optimum performance. In this work we set the thresholds experimentally for each SNR. The second challenge is to estimate the frame energy when there is background noise. Although there are many algorithms in the literature to estimate the frame energy, we used an energy subtraction based algorithm to keep the computational complexity low. If the noisy frame energy is $P_y$ and the average noise power is $P_n$, the estimated clean frame energy is calculated as

$$\hat{P}_s = P_y - \alpha P_n.$$  (1)

The $\alpha$ factor is set experimentally.

One problem with ignoring the low energy speech segments is that, some parts of the phonemes are lost. In the case of a nasal or a fricative the whole phoneme might be lost. This creates a mismatch between the trained words and decoded words. To avoid this mismatch we applied the same thresholding to training files. This ended up decreasing the performance at high SNRs. The problem can be solved by using multiple training templates, and choosing the template based on the average SNR.

4. EXPERIMENTS

4.1. Setup

We used a word based recognizer developed at Mississippi State University as our baseline recognizer [3]. Words are modeled with a 10 state left-to-right Hidden Markov Models (HMM) using 16 Gaussian mixtures per state.

The noise sources we used are from the NOISEX-92 database. We used babble noise. During our discussions with other researchers in the field we noticed that different groups use different algorithms for SNR calculations. We believe this point is crucial for the comparison of performance between groups. We used ITU-T P.56 standard to calculate the average speech energy.

We used TIDIGIT database. The threshold values are set for the TIDIGIT database, which contains typical human speech in normal environments. These thresholds may be adjusted if required by the application.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>0 dB</th>
<th>5 dB</th>
<th>10 dB</th>
<th>15 dB</th>
<th>20 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e^{10}$</td>
<td>93.4</td>
<td>67.6</td>
<td>85.8</td>
<td>92.8</td>
<td>95.9</td>
</tr>
<tr>
<td>$e^{11}$</td>
<td>89.8</td>
<td>81.7</td>
<td>91.7</td>
<td>96.2</td>
<td>97.2</td>
</tr>
<tr>
<td>$e^{14}$</td>
<td>62.5</td>
<td>81.5</td>
<td>89.2</td>
<td>93.1</td>
<td>94.8</td>
</tr>
<tr>
<td>$e^{16}$</td>
<td>66.1</td>
<td>80.0</td>
<td>87.1</td>
<td>89.4</td>
<td>91.4</td>
</tr>
</tbody>
</table>

Table 1. Upper bounds for word accuracy rates at different threshold and SNR levels. Upper bound of the performance is found by extracting the threshold information from clean speech, while using the MFCC parameters from noisy speech.

![Fig. 4](image-url)  
**Fig. 4.** Comparison of performance for missing frame, NLSS and CMS. The unprocessed speech case is also provided for comparison.

4.2. Results and Discussion

Table 1 shows the result when the thresholding is applied on the clean data, but the MFCC features are extracted from the noisy data. Thus, these results are the upper limits for the performance of our algorithm. The significant improvement compared to unprocessed case is clear from the results shown in Table 5. For example, the original performance of 88 percent word accuracy rate (WAR) at 0 dB, improved to 96 for a threshold of $e^{10}$.

Tables 2, 3, and 4 show the actual results with the proposed thresholding algorithms. Although, results are below the upper limit particularly for low SNRs, a significant improvement compared to unprocessed case is clear. Furthermore, the proposed system significantly outperforms results when speech is enhanced with nonlinear spectral subtraction (NLSS), or Cepstral Mean Normalization (CMN) is applied. These two algorithms are used in comparison because of their comparable computational complexities and effectiveness in increasing the performance. The subtraction factor $\alpha = 0.75$, consistently gave better results.

Figure 4 shows the comparison between the upper limit of the performance, the performance with the current
thresholding algorithm, the unprocessed case, and the processed case with NLSS and CMN algorithms. The data points for the upper limit and the proposed system have been chosen from the best performing systems for each SNR. Therefore, we propose having multiple sets and utilizing the set that has the best performance at a given SNR. We assume that the frame SNR is known. Since we do not make any explicit assumptions about the noise types, the 4 sets with different thresholds can be used for any type of noise. At 0 dB and 5 dB the proposed system clearly outperforms the other cases. Also, threshold of $exp(12)$ outperforms other thresholds especially at higher SNRs.

### 4.3. Computational Complexity

The proposed system adds a negligible amount of computational complexity at the front of the ASR system. This additional burden is due to thresholding algorithm. We used a computationally simple thresholding algorithm in this work. The main computational burden is on the Viterbi search module of the system. In our experiments, we observed that our system works significantly faster than the standard system. The decoding speed increased by approximately two folds.

The reason behind this result is the fact that the statistics for portions of the speech that are above the energy threshold (high energy portions) are very different from the portions that are below the threshold (low energy portions) since we set the latter to silence level. Thus, paths that correspond to high energy states of the word templates are pruned very easily if the actual features correspond to silence periods and vice versa. This effect ended up decreasing the average number of active paths at a given time, which in turn increased the search speed.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>0 dB</th>
<th>5 dB</th>
<th>10 dB</th>
<th>15 dB</th>
<th>20 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e^{10}$</td>
<td>37.4</td>
<td>69.3</td>
<td>82.2</td>
<td>89.5</td>
<td>94.7</td>
</tr>
<tr>
<td>$e^{12}$</td>
<td>44.6</td>
<td>79.5</td>
<td>91.2</td>
<td>96.6</td>
<td>98.1</td>
</tr>
<tr>
<td>$e^{14}$</td>
<td>42.3</td>
<td>78.9</td>
<td>90.8</td>
<td>94.6</td>
<td>95.3</td>
</tr>
<tr>
<td>$e^{16}$</td>
<td>41.6</td>
<td>74.8</td>
<td>86.1</td>
<td>89</td>
<td>89.8</td>
</tr>
</tbody>
</table>

Table 2. Word accuracy rates at different threshold and SNR levels for $\alpha = 1.0$.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>0 dB</th>
<th>5 dB</th>
<th>10 dB</th>
<th>15 dB</th>
<th>20 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e^{10}$</td>
<td>47.3</td>
<td>72.2</td>
<td>83.9</td>
<td>91.5</td>
<td>94.9</td>
</tr>
<tr>
<td>$e^{12}$</td>
<td>54.6</td>
<td>81.8</td>
<td>89.9</td>
<td>94.7</td>
<td>95.8</td>
</tr>
<tr>
<td>$e^{14}$</td>
<td>56.3</td>
<td>79.2</td>
<td>89.6</td>
<td>93.5</td>
<td>94.8</td>
</tr>
<tr>
<td>$e^{16}$</td>
<td>52.6</td>
<td>76.8</td>
<td>83.6</td>
<td>88.9</td>
<td>89.9</td>
</tr>
</tbody>
</table>

Table 3. Word accuracy rates at different threshold and SNR levels for $\alpha = 0.75$.

### 5. CONCLUSION AND FUTURE WORK

The missing frame algorithm presented in this paper substantially improves the performance at low SNRs. Furthermore, we observed that the average decoding speed increased by approximately two folds due to effective path pruning at the searching stage. Moreover, no assumption about the noise type is made. This makes the algorithm particularly suitable for real time applications that require low computational complexity and typically work in nonstationary background environments. The upper bound of performance that is shown in Figure 4, shows that the performance can be improved even further using a more effective frame energy estimator. In our future work, we will focus on developing a low complexity energy estimator to improve the performance of the system.

The proposed system uses word based templates, which makes it impractical for a large vocabulary tasks. Furthermore, two different words might end up having the same structure after setting the low energy portions to silence level. We will try to solve these problems in order to use this same idea for a larger vocabulary noisy speech recognition task.

### 6. REFERENCES

