SIGNAL MODELING FOR ISOLATED WORD RECOGNITION

Montri Karnjanadecha and Stephen A. Zahorian

Department of Electrical and Computer Engineering
Old Dominion University
Norfolk, VA 23529, USA

ABSTRACT

This paper presents speech signal modeling techniques which are well suited to high performance and robust isolated word recognition. Speech is encoded by a discrete cosine transform of its spectra, after several preprocessing steps. Temporal information is then also explicitly encoded into the feature set. We present a new technique for incorporating this temporal information as a function of temporal position within each word. We tested features computed with this method using an alphabet recognition task based on the ISOLET database. The HTK toolkit was used to implement the isolated word recognizer with whole word HMM models. The best result obtained based on 50 features and speaker independent alphabet recognition was 98.0%. Gaussian noise was added to the original speech to simulate a noisy environment. We achieved a recognition accuracy of 95.8% at a SNR of 15 dB. We also tested our recognizer with simulated telephone quality speech by adding noise and band limiting the original speech. For this "telephone" speech, our recognizer achieved 89.6% recognition accuracy. The recognizer was also tested in a speaker dependent mode, resulting in 97.4% accuracy on test data.

1. INTRODUCTION

Continuous speech recognition systems have been developed for many real-world applications, often using commercial low-cost speech recognition software. However, high performance and robust isolated word recognition, particularly for the letters of the alphabet recognizer and for digits, is still useful for many applications such as recognizing telephone numbers, spelled names and address, and ZIP codes.

Because of the potential applications, as mentioned above, many isolated word recognizers are optimized for the digits or alphabet or both (alphadigit). The alphabet recognition task is particularly difficult because there are many highly confusable letters in the alphabet set---for example the great acoustic similarity among the letters of the E-set (b, c, d, e, g, p, t, v, z) or for the (m,n) pair. Also, since language models cannot generally be used, the alphabet recognition task is a small, challenging, and potentially useful problem for evaluating acoustic signal modeling and word recognition methods

Several techniques have been proposed to improve isolated word recognition systems. For example, the best result in a speaker independent alphabet recognition was obtained using a multi-tier phoneme-based Hidden Markov Model (HMM) recognizer [5]. Disadvantages of phoneme-based HMM recognizers are the system complexity and the phonetic transcription of the training words has to be known.

The main contribution of this paper is to present a method for isolated word recognition which is easier to implement than the state of the art systems introduced to date, and one which gives better performance than any of these previously introduced systems.

The ISOLET database, [1], was used for all experiments reported in this paper. This LDC distributed database was intended for evaluation of isolated word recognizers and it has therefore been used by many researchers. Thus, it is possible to directly compare results. Most of the experiments were done in a speaker independent fashion using all files from the database, i.e., 120 speakers (60 male and 60 females) for training and 60 speakers for testing (30males and 30 females).

The HTK toolkit [6], originally developed at Cambridge university, and now distributed by Entropics Inc., was used to implement the HMM recognizer. In our lab, we ported this toolkit to Windows NT, and conducted all experiments under this operating system.

In [4], which appeared in November 1998, we achieved a best result of 97.3% for speaker independent alphabet recognition. With the modifications introduced in this paper we achieve even better performance, i.e. 98.0%. This represents a reduction in error rate of about 25%, or with 1560 test tokens, the number of error tokens was reduced to 31 from 42.

This paper is organized as follows. Section 2 describes the signal modeling procedures. Experimental details and result discussions are presented in Section 3 and conclusions are drawn in Section 4.

2. SIGNAL MODELING

Our feature extraction method, which has been shown to work well with several classification and recognition problems, has been under refinement for a very long time. The method presented in this paper is a variation of the method used in [4] and [7]. In this paper, we summarize the method, including the changes which resulted in the error rate reduction.

First, after second order pre-emphasis with a pre-filter centered at 3200Hz, Kaiser-windowed 20-ms speech frames were analyzed with a 512-point FFT every 5 ms. A Kaiser window beta of 8, that is a window slightly "smoother" than a Hamming window, was used. The spectral range was limited to 60 dB for each frame, using a floor. 10 modified cosine terms over

frequency were computed for each spectral frame. These 10 terms, very similar to cepstral coefficients, were computed with a bilinear warping factor of .45 over the frequency range of 60 Hz to 7600 Hz. These 10 terms in turn were each represented by a 5 term modified cosine expansion over time, using a "block" window with variable length. Thus each block was represented by 50 spectral/temporal features, as given by the following equations.

$$DCSC(i,j) = \int_{-0.5}^{0.5} DCTC(i,t)\Theta_j(t)dt \qquad (1)$$

$$\Theta_j(t) = \cos[\pi j h(t)]\frac{dh}{dt} \qquad (2)$$

In equation 1, the DCTC terms ($0 \le i \le 9$) are the cosine terms computed over frequency for each frame. The DCSC terms ($0 \le j \le 4$) are the terms which represent the trajectory information. Equation 2 is the method for computing the basis vectors over time (with *h(t)* a Kaiser window function). Both of these equations are in terms of "normalized" time *t* ($-0.5 \le t \le 0.5$). In the next few paragraphs, we describe the method used to determine the actual time on which segment features are based, which we call the block length, or number of frames used to compute each set of 50 features.

This block length was adapted to the position within the utterance. In particular, at the beginning of an analyzed token, a block size of 6 frames (i.e., a 45 ms total duration, including end effects of the analysis frames) was used. As the analysis window moved forward, the block size increased until a specified maximum was reached. The block size was then fixed at this maximum until the end region of utterance was reached. At this point the block length was again gradually reduced until, for the very final block, it again reached 6 frames. Time "warping" was also applied to each block, again using a Kaiser window, but with a beta value of 5.0 for the maximum length blocks. The Kaiser window beta also varied from 0 for the 45 ms blocks up to 5.0 for the maximum block length. Thus, the features gave better time resolution for the onset and offset portions of each word, and less time resolution in the central portions of each word. The block features were re-computed every 10 ms. No manual segmentation or phonetic labeling was required or used. The primary modification, relative to [4] is that the block length was varied at both ends of each analyzed utterance, rather than only for the beginning section.

As an example, suppose block features are to be computed from a speech token having N analysis frames, with block lengths ranging from 1 to 5 frames per block. For this example, the block processing starts with a minimum block length of 1 frame, increasing to a maximum block length of 5 frames and used a block advance rate of 2 frames. Figure 1 illustrates the increasing block length at the beginning of the analysis token. As can be seen, the block length starts with 1 frame then increases in size to 3 and 5 frames consecutively. The block length of 5 frames is fixed until the end of token is almost reached. At this position, the block length is reduced from 5 frames to 3 frames then to 1 frame as shown in Figure 2.

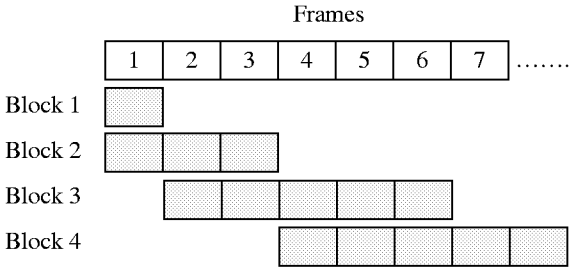

Figure 1. The increase of block length at the beginning of an utterance.

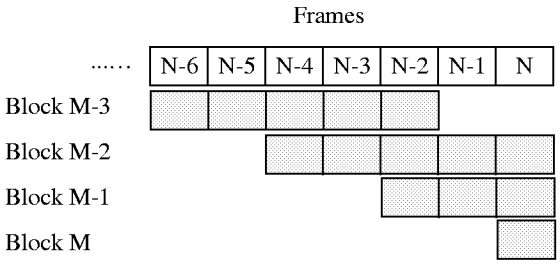

Figure 2. The decrease of block length at the end of an utterance.

Note that the warping factor over time is adjusting according to the block length. For example, the factor is zero when the block length is minimum and the warping factor is 5 at the maximum block length. The idea behind the adjustable warping factor is that for a shorter block, each frame gives equal contributions. For a longer block the frames in the middle of the block gives more contribution (i.e., more emphasized).

3. EXPERIMENTS, RESULTS AND DISCUSSION

The ISOLET database from OGI [1] was used in all experiments. The database is comprised of the English alphabet letters spoken by 150 speakers. There are 75 male and 75 female speakers. Each speaker uttered the same word twice. Thus there are totally 7800 utterances. The database is divided into 5 groups: ISOLET-1, ISOLET-2, ISOLET-3, ISOLET-4, and ISOLET-5. Each group has an equal number of speakers. Utterances were recorded as isolated words with the sample frequency of 16000Hz and a 16-bit A-to-D system. The speech signal-to-noise ratio (SNR) reported by OGI is 31.5 dB with a standard deviation of 5.6 dB.

Although all speech files were reasonably accurately endpointed in the original distribution list, in our previous work, we found a small but significant improvement in recognition performance using data with endpoints refined by the modified endpoint detection algorithm of [3]. Therefore, for all experiments reported in this paper, we used the same endpoint detected version of ISOLET database as was used in [4].

For all experiments presented in this paper, the HTK toolkit was used to implement a word-based HMM recognizer. In each

experiment, there were 26 HMM models trained to recognize all 26 English alphabets. Each model had 5 states and 3 multivariate Gaussian mixtures with a full covariance matrix. In the training phase of each alphabet, every training utterance was segmented into equal lengths and then initial model parameters were estimated. Note that no Baum-Welch iterations were done, as there were not found to be useful. Next, the Viterbi decoding algorithm was applied to determine an optimum state sequence of each training token. Every token was re-segmented based on its corresponding optimum state sequence. Model parameters were re-estimated repeatedly until the estimates were unchanged or the maximum number of iterations were reached. Again the Viterbi algorithm was applied in the testing phase to determine the most likely model that best matched each test utterance.

3.1 Experiment I

The purpose of this experiment was to evaluate our feature extraction technique in a speaker independent alphabet recognition task. The training data was comprised of all utterances from ISOLET-1 to ISOLET-4 (i.e. totally 6240 tokens) and test data was comprised of all tokens from ISOLET-5 (totally 1560 tokens). The feature analysis together with HTK toolkit described above were used. The minimum block length was fixed at 6 frames while the maximum block length was varied from 6 to 100 frames. The table below shows recognition results with various maximum block sizes.

Table 1. Recognition accuracy with various maximum block length.

Maximum block length (frames)	Recognition accuracy (%)
6	94.2
10	95.0
20	96.9
30	97.2
40	98.0
50	97.4
60	97.6
70	97.4
80	97.2
90	97.1
100	97.3

The best performance was obtained with a maximum block length of 40 frames which is equal to the time duration of 215 ms. With these best parameters, the recognition accuracy of each alphabet subset is shown as in the following confusion matrix. Note that rows and columns with no error entries are not shown.

Table 2. Error section of confusion matrix corresponding to the best test result (2% error).

	B	C	D	E	F	G	I	J	L	M	N	O	P	Q	R	T	U	V	W	Y	Z
B	58	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.
C	.	59	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1
D	.	.	59	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
E	1	.	.	59	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
F	.	.	.	.	59	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.
G	.	.	.	.	.	59	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.
L	.	.	.	.	.	.	.	59	.	1	.	.	.	.	.	.	.	.	.	.	.
M	.	.	.	.	.	.	.	.	58	2	.	.	.	.	.	.	.	.	.	.	.
N	.	.	.	.	.	.	.	.	7	52	.	.	.	.	.	.	.	1	.	.	
O	.	.	.	.	.	.	.	1	.	.	59	.	.	.	.	.	.	.	.	.	.
P	2	.	.	.	.	.	.	.	.	.	.	58	.	.	.	.	.	.	.	.	.
R	.	.	.	.	.	.	.	.	1	.	.	.	.	59	.	.	.	.	.	.	.
T	.	.	.	.	.	.	1	.	.	.	.	2	.	.	57	.	.	.	.	.	.
U	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	59	.	1	.	.
Y	.	.	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	59	.
Z	.	3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	56

This recognition accuracy with respect to each alphabet set can be described as follows. The recognition rate of alphabet letters in the E-set is 97.2%, the recognition rate of the (m,n) pair is 91.7%, and recognition rate of the remaining letters is 99.3%. Note that the poorest performance is obtained from (m,n) pair. The result shown in the table is a 25% error reduction compare to the best result reported in [4] and is the highest result ever reported on this test set.

The accuracy of 98.0% corresponds to 31 error tokens, pronounced by 14 speakers out of the 60 test speakers (with 12 errors obtained from just 2 speakers). The number of errors was small enough to be inspected graphically and by listening. After listening to all of the error tokens, we concluded that there are 4 situations for which tokens were misrecognized. There were 9 tokens which appeared to have an endpoint detection problem. We believe that if endpoints were more accurately obtained, these errors would be eliminated or reduced. There are 8 tokens that were pronounced in an unusual way, all by single speaker. Also there were 8 tokens that were misrecognized because they are so similar to other letters, that, even with careful listening, they were difficult to recognize. Finally, there were 6 tokens that sounded reasonably clear and understandable, and there was no reason that could be specified for the error in machine performance.

Note that the best parameter values obtained from this experiment were also used in later experiments.

3.2 Experiment II

This experiment was conducted to determine the overall performance of the system. The test set was rotated to avoid unfair tuning of parameters. As stated above, the ISOLET database contains 5 subsets. For this experiment, each subset was used once as a test set while the rest were used as the training set. Results are depicted in Table 3.

Table 3. Results obtained with various test sets.

Test set	Recognition accuracy (%)
ISOLET-1	97.9
ISOLET-2	97.4
ISOLET-3	97.4
ISOLET-4	97.4
ISOLET-5	98.0
	Average = 97.6

The average performance of 97.6% is achieved in this experiment which is shown that our recognizer works very well with any test set and confirmed that it has not been tuned to perform well only on a particular test set.

3.3 Experiment III

The purpose of this experiment was to investigate the recognition performance on noisy and band-limited speech. The best parameter values obtained from the previous experiment were used with noisy and band-limited speech. Gaussian noise was added to all speech tokens to have a SNR of 15 dB. Band-limiting was achieved by selecting frequency components that are in the desired range. For example, in this experiment speech data was band-limited to the telephone channel bandwidth, i.e. 300-3200 Hz. Gaussian noise was also added to band-limited speech in order to simulate telephone speech. Results are shown in the following table.

Table 4. Results with noisy and band-limited speech

Frequency range (Hz)	SNR (dB)	Recognition accuracy (%)
60-7600	clean speech	98.0
60-7600	15	95.8
300-3200	clean speech	92.9
300-3200	15	89.6

The recognizer performed very well with noisy speech and band-limited speech. The most interesting part is a recognition accuracy of 89.6% was achieved in the case of simulated telephone speech. This does not guarantee that performance will be the same for real telephone speech. Although telephone speech cannot be modeled by only band-limiting and additive Gaussian noise, the noise level used in our tests (15 dB SNR) is much higher than would be typically found on a telephone.

3.3 Experiment IV

This experiment was intended to show the system performance on a speaker dependent alphabet recognition task. The database was organized in multi-speaker configuration. All first utterances from ISOLET-1 to ISOLET-4 were used for training and all second utterances from the same subsets were used for testing. In the other words, totally 120 speakers were used for training and testing. Recognition accuracy of test data obtained from this experiment was 97.4%. This is favorable to the result reported in [2]. Note, however, that the result in this experiment is slightly lower than that of the speaker independent case. Presumably, the reduction in the number of training tokens was more detrimental, than any increase due to the speaker dependent effects.

4. CONCLUSIONS

Word-based HMM recognizers have been claimed to give poorer recognition performance compared to phoneme-based recognizers as presented in [5]. Experimental results reported in this paper have shown that if features are computed properly, whole word recognizers can, in fact, surpass the best reported results for phoneme based recognizers. Whole word HMM based recognizers are easier to implement, at least for isolated word recognition. Results show that the proposed signal modeling techniques are straightforward, efficient and robust. The technique can also be extended to work with continuous speech. Block length can be varied depending on "rate of change of the spectrum" with shorter block lengths used in sections of high spectral derivative and longer block length used in sections with low spectral derivative.

5. REFERENCES

[1] Cole, R., Muthusamy, Y., and Fanty, M., "The ISOLET spoken letter database," Tect. Rep. 90-004, Oregon Graduate Inst., 1990.

[2] Cole, R., Fanty, M., and Muthusamy, Y., "Speaker-independent recognition of spoken English letters," *in Proc. Int. Joint Conf. Neural Netowrks,* Vol.2 June 1990, pp.45-51.

[3] Dermatas, E. S., Fakotakis, N. D., and Kokkinakis, G. K., "Fast Endpoint Detection Algorithm for Isolated word Recognition In Office Environment," *Proc. ICASSP'91, Toronto, Canada, MAY 1991, pp. 733-736*

[4] Karnjanadecha, M., and Zahorian, S. A., "Robust Feature Extraction for Alphabet Recognition," to be appear in *Proc. ICSLP'98, Sydney, Australia, Nov.-Dec. 1998.*

[5] Loizou, P. C., and Spanias, A. S., "High-Performance Alphabet Recognition," *IEEE Trans. Speech and Audio Processing. Vol. 4, no. 6, pp. 430-445, 1996.*

[6] Young, S. J., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., *Hidden Markov Model Toolkit V2.1 reference manual*, Technical report, Speech group, Cambridge University Engineering Department, March 1997.

[7] Zahorian, S. A., Silsbee, P. L., and Wang, X., "Phone Classification with Segmental Features and a Binary-Pair Partitioned Neural Network Classifier," *Proc. ICASSP97, Munich, Germany, Apil. 1997, pp. 1011-1014.*