NOISE ROBUST SEGMENT-BASED WORD RECOGNITION USING VECTOR QUANTISATION

Ramalingam Hariharan, Juha Häkkinen, Kari Laurila and Janne Suontausta
Speech and Audio Systems Laboratory, Nokia Research Center
P.O. Box 100, FIN-33721 Tampere, Finland.
E-mail: {ramalingam.hariharan, juha.hakkinen, kari.laurila, janne.suontausta}@research.nokia.com
FAX: +358 3 272 5897

ABSTRACT

Segment-based speech recognition systems have been proposed in recent years to overcome some of the deficiencies of the current state-of-the-art HMM based systems. In this paper, we present a segmental speech recogniser, where the speech trajectory segments are modelled using their mean, variance and shape. The shape is chosen from a codebook of global vector quantised trajectories, obtained from uniformly segmented training utterances. Experiments were done for a speaker dependent isolated word recognition application under different noise environments. The results have shown that this segment based approach outperforms HMM based speech recognition systems under similar test conditions. In adverse noise conditions, up to 34% error rate reduction was achieved.

1. INTRODUCTION

Most of the state-of-the-art speech recognition systems are based on Hidden Markov Models (HMMs) for acoustic modelling. However, there are several shortcomings in the use of HMMs. They assume statistical independence and identical distribution of observation in each state. Weak duration modelling is another major constraint in this type of approach. Many researchers have adopted the segment based approach to model the temporal variations within states. In segmental modelling, speech parameters are represented by trajectories, i.e. sequences of points in the parameter space. Ostendorf et al. [1] developed the idea of using segmental information with variable-length segments and stochastic models of segment parameters for phoneme-based continuous speech recognition. Siohan et al. [2] used stochastic trajectory modelling, where speech units are modelled as mixtures of state sequences. Some researchers have used parametric functions to model the trajectories [3]. In addition to the pure segmental methods, several approaches where segmental information is incorporated to the HMM structure have also been proposed [4].

In this work we describe a new segment based approach using a vector quantised codebook of feature vector trajectories. The speech trajectories are characterised by the mean, variance and shape of the particular segment. The shape is a vector chosen from a set of universal codebook trajectories, generated from uniform segments of training utterances. The vector quantised codebook trajectories can model the word models better than accurate modelling with the exact trajectory shape, which suffers from insufficient data. This has been verified and described later in this paper.

The remainder of the paper is organised as follows. The next section describes the creation of word models and the training procedure. The recognition methodology is outlined in detail in section 3. Finally, the recognition results obtained using the segmental approach are presented in section 4. Recognition results using an HMM based system have also been presented for comparison.

2. WORD MODEL AND TRAINING

In segmental modelling, speech is modelled as a sequence of feature vectors in the parameter space, i.e. in the cepstral domain, and it is assumed to consist of consecutive segments [1]. Ideally, each segment would correspond to an acoustic event, but here we take a simpler approach where the length of the segments is fixed during the training of the word models. We use the term segmental feature vector for a speech segment. Each component of the segmental feature vector represents the time trajectory of the corresponding cepstral coefficient.

Our word model is a description of the time trajectories of cepstral coefficients for a given training utterance. The word model has states, each of which corresponds to a segmental feature vector. In one state, the time trajectory of a segmental feature vector element is characterised by three parameters: the mean, the standard deviation, and the shape. The shape is obtained from a given codebook. Thus, one word model is composed of the following parameters

$$\left\{\mathbf{W}_f^j, \mu_f^j, \sigma_f^j\right\}, \tag{1}$$

where $\mathbf{W}_f^j, \mu_f^j$ and σ_f^j are the code vector, mean and standard deviation of the *f*th feature vector element in the *j*th state of the word model.

The means of the elements of the segmental feature vectors can differ widely. In addition, the elements of the segmental feature vectors may exhibit the same pattern of variation, but the gain can be different. Thus, by computing the mean and variance of the segmental feature vector element, and normalising the element to have zero mean and variance of one, some elements of the feature vectors become quite similar and the different shapes can be characterised by a smaller set of vectors.

2.1. Training of the word model

In the training phase, we estimate the parameters of the word models.

The feature vector elements are segmentally normalised as described in [5], so the mean and variance of every cepstral coefficient are close to zero and one. The speech part of the utterance is segmented uniformly into segments with the given number of frames, and in this way the segmental feature vectors for the training utterance are generated.

In each state of the word model, the means and variances can be estimated from the segmental feature vectors of the training utterance. In shape estimation, the mean of the segmental feature vector elements is removed. After that the elements are divided by their standard deviations. The resulting vector is compared against the code vectors, and the code vector which is closest according to the Euclidean distance is our shape parameter. Figure 1 illustrates the fit between the word model obtained with our training method and the training utterance 'Päivi'.

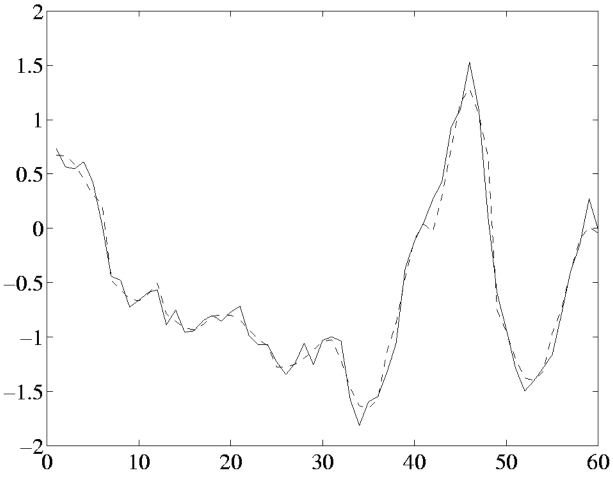

Figure 1. The fit between the first feature vector element of the utterance 'Päivi' (solid line) and the reconstructed trajectory (broken line).

2.2. Training of the codebook

The codebook should be an adequate description of the different possible shapes of the segmental feature vector elements. For codebook training, we have a set of end-pointed utterances for which the feature vectors have been calculated. The training utterances are spoken by various speakers, so the codebook is trained on a general data set. The speech parts of the utterances are segmented uniformly and so we obtain the segmental feature vectors from the training utterances.

The mean and standard deviation of the segmental feature elements are normalised to zero and one, respectively, as it is done when training the word models.

After the scaling, the clusters of the different segmental feature vector elements are determined. The cluster is described by its centroid and by the nearest neighbour rule. The centroid represents the typical feature vector trajectory within the cluster. The nearest neighbour rule assigns each segmental feature vector elements to the cluster, the centroid of which is closest to the element.

In our system, the K-means (Lloyd) algorithm is used to find the cluster centroids. Our code vectors are the centroids of the clusters. In Figure 2, the codebook used in training the model shown in Figure 1 is presented. The codebook has ten code vectors and the segment length is six frames.

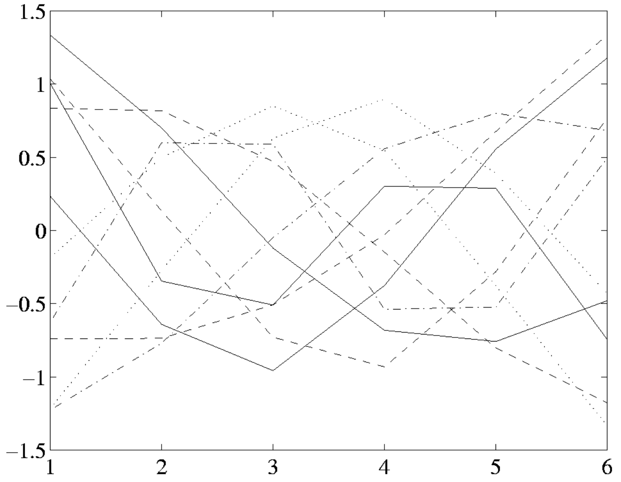

Figure 2. The codebook with ten code vectors, segment length is six frames.

3. RECOGNITION METHOD

The proposed segment-based speech recogniser finds the cumulative distance of each word model and the optimally warped (within constraints) input utterance. The Viterbi algorithm is used for finding the best segmentation and the corresponding cumulative distance.

Garbage models are used to model the non-speech regions before and after the actual test utterance.

3.1. Reconstruction of the model trajectory

Before the distance between the input segment trajectory and the word model segment can be computed, the word model trajectory must be reconstructed from the model parameters according to

$$\mathbf{w}_f^j = \mu_f^j + \sigma_f^j \cdot \mathbf{W}_f^j , \qquad (2)$$

where $\mathbf{w}_f^j$ is the reconstructed segmental feature vector element *f* in the segment *j*.

3.2. Time re-scaling of the trajectory

Input trajectories of different lengths must be warped in time domain to compare them with the fixed-length model segments. We use the time sampling approach described in [1]: the new sample points are calculated according to the warping and the corresponding sample values are interpolated from the existing ones.

The N_0 new sample locations in the *N*-length trajectory segment of the *f*th feature vector component,

$$Y_f^N(n) \overset{\Delta}{=} \left[y(n-N+1), y(n-N+2), \ldots, y(n) \right],$$

are given by

$$n_i = n - N + 1 + i\frac{N-1}{N_0 - 1}, \quad i = 0,1,\ldots,N_0 - 1. \quad (3)$$

Three different deterministic interpolation methods for finding the new sample values $\hat{Y}_f^{N_0}(n)$, are compared: 0th order (sample-and-hold), 1st order (linear) and cubic interpolation.

3.3. Distance calculation

We use the squared Euclidean distance between the warped input segment and the reconstructed word model segment.

The distance between the *j*th segment of the current word model and the warped segment (originally of length *N*) is defined as

$$d_j = \frac{N}{F \cdot N_0} \sum_{f=0}^{F-1} \sum_{i=0}^{N_0-1} \left[w_f^j(i) - \hat{y}_f(i) \right]^2, \quad (4)$$

where *F* is the number of feature vector elements.

It can be noted that the described distance calculation scheme is effectively the same as Gaussian probability calculation in the log-domain using unity variances.

3.4. Delta warping compensation

The reasoning behind the compensation scheme is that when the test utterance is spoken at a different rate from the training utterance, the amplitude of the delta coefficients changes (faster changes increase derivative values). So, when the Viterbi decoder evaluates different warpings for the input segments it should, in addition to changing the temporal length of individual segments of delta coefficients, scale their amplitudes as well.

The proposed method is realised by scaling the delta components of the word models before comparison with the input trajectory. We define the scaling coefficient as

$$sc = \sqrt{\frac{N_0}{N}}, \quad (5)$$

where *N* is the length of the input segment and N_0 the length of the model segment. The square root operation is a heuristic way of ensuring that too much scaling is not applied.

4. EXPERIMENTS

We conducted several recognition experiments to verify our approach for speaker-dependent isolated word recognition. We used two different databases containing four (DB1) and six (DB2) male speakers. In DB1, a set of 30 short (duration less than one second) names was spoken 12 times by each speaker. In DB2, each speaker had his own set of 30 short names repeated 12 times. The first utterance of each set of the same name was used for training, the rest for testing, totalling 330 test utterances per speaker. Single utterance was used for training of each word model. We conducted the tests in several signal-to-noise ratios. The noise (recorded car noise) was added to the test utterances during the tests.

We used segment length 6 (the duration minimum and maximums were 4 and 9, respectively) universally in all tests. The use of longer segments increases computational complexity and doesn't fit well in the codebook framework, as our simulations also indicated. We used 13 MFCCs and their delta values, normalised as described in [5], as the input to the recogniser. Delta warping compensation and linear interpolation were used, unless otherwise mentioned. All results are reported in percentages of correctly recognised words.

4.1. Comparison of the different segmental modelling approaches

We did several tests with different segmental modelling parameters to find their effect on recognition rates. These tests were performed only for DB1. Table 1 shows the effect of delta warping compensation. There is a clear improvement in performance, especially at low SNRs. Table 2 illustrates the effect of codebook size on the recognition rate. As we expected, the most accurate method, where the exact trajectory is used as the model, suffers from inadequate amount of training data. The performance of the interpolation methods, as shown in Table 3, is quite similar, linear being the best. Even the use of zeroth order (sample-and-hold) interpolation could be justified because of the simpler implementation involved.

Table 1: The effect of delta warping compensation on recognition results. 30 code vectors were used in the test.

SNR \ compens.	OFF	ON
clean	98.71	99.17
0 dB	97.05	97.95
-10 dB	92.42	94.54

Table 2: The effect of modelling accuracy (codebook size) on recognition results. 'Exact' means that no vector quantisation was used, but the trajectory was stored frame by frame.

SNR\CB	10	30	90	exact
clean	99.17	99.17	99.17	99.25
0 dB	97.65	97.95	98.11	97.96
-10 dB	93.86	94.54	94.39	93.03

Table 3: The effect of interpolation on recognition results.

SNR \ Interp	0th order	1st order	3rd order
clean	99.17	99.17	99.17
0 dB	97.73	97.95	97.81
-10 dB	93.94	94.54	94.32

4.2. Segmental modelling vs. HMMs

Finally, we compared our best segment-based recogniser with an HMM-based recogniser. The HMM-based recogniser used the same segment and duration parameters and the same garbage model as our approach. Tables 4 and 5 show the recognition results of the HMM-based and the segment-based methods for the two databases. The proposed segment-based recogniser achieves up to 34% error rate reduction over the HMMs at -10 dB SNR.

Table 4: Full results for the HMM and segment model based recognisers (DB1).

SNR	HMM	Segmental (CB=30)	Error rate reduction
clean	98.34	99.17	49.8
5 dB	97.20	97.88	24.4
0 dB	96.97	97.96	32.5
-5 dB	96.36	96.90	14.6
-10 dB	93.48	94.55	16.3

Table 5: Full results for the HMM and segment model based recognisers (DB2).

SNR	HMM	Segmental (CB=30)	Error rate reduction
clean	99.85	99.80	-32.0
5 dB	99.49	99.45	-9.8
0 dB	98.69	99.01	24.5
-5 dB	97.53	98.44	36.8
-10 dB	95.00	96.70	34.0

5. CONCLUSIONS

We presented a novel approach to segment-based speech recognition using a vector quantised codebook of feature vector trajectories. We showed that even a moderately small codebook effectively captures the shape of feature vector trajectories.

We demonstrated the performance of our approach in a comparison with an HMM-based recogniser. We achieved up to 34% error rate reduction in adverse noise conditions. However, the computational cost of segment-based recognition is probably not justified by the performance improvement in the simple speaker dependent isolated word recognition task. In the future, we are planning to apply the same framework to speaker independent speech recognition, where segmental modelling approach has been reported to perform much better.

REFERENCES

[1] Ostendorf M., Digalakis V.V., and Kimball O.A., "*From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition*," IEEE trans. on ASSP, vol. 4, no. 5, September 1996, pp. 360-378.

[2] Siohan O., and Gong Y., "*A semi-continuous stochastic trajectory model for phoneme-based continuous speech recognition*," In Proc. ICASSP, Atlanta, USA, May 1996, vol. 1, pp. 471-474.

[3] Liu C., Wang H., "*A Segmental Probabilistic model of speech using an orthogonal polynomial representation: Application to text-independent speaker verification.*" Speech Communication, vol. 18, 1996, pp. 291-304.

[4] Holmes J.W, Russell J.M., "*Experimental evaluation of Segmental HMMs*", in Proc. ICASSP, Detroit, USA, May 1995, pp. 536-539.

[5] Viikki O., Laurila K., "*Noise robust HMM-based speech recognition using segmental cepstral feature vector normalisation*," in Proc. ESCA-NATO, Pont-à-Mousson, France, April 1997, pp. 107-110.