

DYNAMIC FEATURES FOR SEGMENTAL SPEECH RECOGNITION.

Naomi Harte Saeed Vaseghi Ben Milner*

School of Electrical Engineering and Computer Science, The Queen's University of Belfast, Belfast, N.Ireland.

*British Telecom Research Laboratories.

n.harte@qub.ac.uk

ABSTRACT

Speech models and features that emphasise the dynamic aspects of speech can provide improved speech recognition. The cepstral time matrix has been established as a successful method of encoding dynamics. This paper extends this set of dynamic features, considering cepstral time features on both a segmental and subsegmental level. This offers the potential of using a conditional pdf for the state observation within a HMM and incorporating this into the training stage. Methods of linear discriminative analysis are applied to the new feature set to identify the subset of features making the greatest contribution to the task of recognition.

1. INTRODUCTION

The identification of a robust feature set for recognition is an important element in building a successful recognition system. The cepstrum coefficients have been identified as a reliable feature set and their performance has been enhanced in recent years through the incorporation of dynamic features by the inclusion of first and second order differential features [1]. More recently, the cepstral time matrix has been shown to provide a systematic method for encoding the short term transitional dynamics of a speech sound [2,3]. This is particularly important for transient events in which time variation of speech features can be a significant factor in accurate modelling and recognition. In this work we deal with a highly confusable vocabulary - the E-Set - which is one such vocabulary where accurate capture of the transitional dynamics of the speech sound can be considered particularly critical to recognition.

Using a filterbank implementation of the cepstral time matrix, the current work goes on to explore the use of dynamic features at two different levels - at a subsegment level and a segmental level. Such a two tier system of dynamic features offers the possibility of further conditioning the state observation in a HMM on the segmental level features and incorporating this into the training stage of a HMM recogniser.

Another important issue in speech recognition has been to identify the best feature subset from a large number of features. In a given feature set, it is unlikely that all the features will contribute equally to the task of recognition and this becomes more true as the feature set grows. Methods of Linear Discriminative Analysis aim to identify the features or combinations of features which are the most important for recognition. In the context of the present work, existing methods

of Linear Discriminative Analysis are applied to the new feature set to explore the potential of these methods of discrimination.

2. DYNAMIC FEATURES

The conventional method for inclusion of speech dynamics is to augment cepstral features with first and second order dynamics. If we denote conventional cepstral features as $c(n,m)$, the m th coefficient at time n , we can express the first order dynamics in the form:

$$\dot{c}(n,m) = \sum_{n=-N}^N w_n c(n,m) \quad (1)$$

and the second order dynamics as:

$$\ddot{c}(n,m) = \sum_{n=-N}^N w_n \dot{c}(n,m) \quad (2)$$

where the weighting function w_n determines the form of dynamics used. For instance, if the weighting scheme $w_{n-1}=1$, $w_n=0$ and $w_{n+1}=1$ is employed, this will yield the first and second order difference features.

An alternative for w_n is the use of the DCT basis functions. The use of these basis functions yields a cepstral time matrix as a representation of first order dynamics. The cepstral time matrix provides a systematic method for the decomposition of transitional dynamics and has demonstrated a number of advantages over differential parameters, including channel robustness and relative ease of adaptation in noise.[4] The work described in this paper also exploits the DCT basis functions in (2) above, to investigate higher order dynamics and the use of cepstral time matrices.

2.1 Filterbank Implementation of Cepstral Time

In previous implementations, the log spectral vectors were obtained from a DFT of a block of speech samples. The length of a speech block (typically 32ms) used in the DFT for conversion of speech from time domain samples to spectral domain samples imposes a fundamental limitation on the time resolution of the cepstral time features. Here a filterbank implementation is employed which provides better time resolution and enables the use of the DCT on longer sequences of cepstral vectors.

Figure 1 shows a block diagram of the implementation used. The DFT based spectra is replaced by a bank of bandpass digital filters. Each bandpass filter was designed using a second order Butterworth filter. The centre frequencies and bandwidth of the 21 bandpass filters were designed to be the same as that of the DFT based mel spaced spectral features.

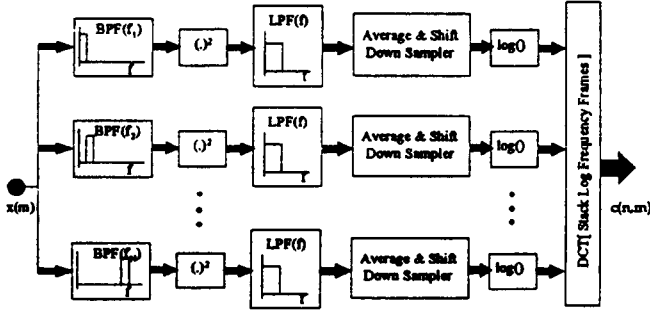


Figure 1 : Filterbank Implementation of Cepstral Time.

The input signal, $x(m)$, is passed through the filterbank and divided into 21 sub-band signals. Each sub-band signal is squared and averaged to extract the variations of the speech spectral envelope. The block "Average, Shift and Down Sample", gives the running average of N samples and then shifts the averaging window by s samples.

After down sampling and logarithmic operation, L spectral vectors are stacked and transformed to cepstral vectors via a DCT. A subsequent DCT yields the cepstral time matrix.

3. SEGMENT AND SUBSEGMENT BASED FEATURES

As part of this work, the use of two different sets of dynamic features was investigated. The conventional method for inclusion of speech dynamics has been to augment cepstral features with first and second order derivatives. Thus, a typical cepstral feature set consists of features at a static level, first order differential and second order differential. We have denoted conventional cepstrum as $c(n,m)$. The use of the cepstral time matrix allows us to encode short term transitional dynamics over a time frame determined by the number of stacked cepstral vectors over which we perform the DCT. Thus, in our cepstral time matrix this DCT yields the k th DCT component of the time variation of the m th cepstral coefficient at time n . We denote this:

$$\dot{c}(n,m,k) = \sum_{p=0}^{T-1} c(p,m) \cos\left(\frac{(2p+1) \cdot k\pi}{2T}\right) \quad (3)$$

where T represents the number of cepstral vectors the DCT is performed over. In general the k th column of the matrix represents the dynamic component varying at a speed of k/T Hz., where T is the actual matrix length in seconds. The zeroth column of the cepstral time matrix represents the d.c. component

of time variations of cepstral coefficients and represents the static features. This column also includes the channel distortions. The current work also incorporated the use of dynamic features on a segmental level. This was done through the inclusion of long-term variations of the cepstral time features. These features can be derived from the first order dynamics as:

$$\tilde{c}(n,m,k) = \sum_{q=n-\frac{(S-1)}{2}}^{n+\frac{(S-1)}{2}} \dot{c}(q,m,k) \cos\left(\frac{(2(q-n+\frac{S-1}{2})+1) \cdot \pi}{2S}\right) \quad (4)$$

This is the long-term variation, over a segment S , of the k th DCT component of the time variation of the m th coefficient at time n . To obtain these features, the preserved columns of the cepstral time matrix were concatenated to form a cepstral time vector. A number of similar vectors were taken forwards and backwards in time and a DCT applied. The first column of this new matrix was preserved to represent the velocity of the successive columns of the cepstral time matrix, over the longer segment, the length of the segment being determined by the number of cepstral matrices over which the variation was taken.

The use of these features provides the possibility of using a conditional pdf for the state observation in a Hidden Markov Model, in which the observation of a subsegmental speech feature vector is conditioned on the segmental or frame features and can be expressed as $f_{x_s|x_f}(x_s|x_f)$.

Experiments were carried out to explore the joint use of the two sets of features; to compare the performance with the original cepstral time matrices and to investigate the effect of taking the segmental features over different segment lengths.

4. LINEAR DISCRIMINATIVE ANALYSIS OF CEPSTRAL TIME

Linear Discriminative Analysis (LDA) seeks to identify the best linear combination of features from a given feature set. The current work sought to apply existing methods of LDA to the new feature set to optimise performance and achieve dimensionality reduction.

An n -dimensional feature space containing feature vector \mathbf{x} can be reduced to an m -dimensional feature space containing \mathbf{y} by applying the n by m linear transform \mathbf{A} where $\mathbf{y}=\mathbf{A}^T\mathbf{x}$. For the purpose of this work, the transform \mathbf{A} was obtained as the eigenvector matrix of the product $\mathbf{W}^{-1}\mathbf{B}$ [5], where \mathbf{W} is the pooled within class covariance matrix and \mathbf{B} the between class covariance matrix. The transform seeks to optimise separability in the sense that the axes of the transformed feature space are aligned on average with the directions of maximum separability. The within class covariance matrix was obtained from HMMs trained with the original training data.

Dimensionality reduction is achieved by choosing the m largest eigenvalues of the diagonalised matrix \mathbf{B} and corresponding m eigenvectors to form the matrix \mathbf{A} . In this way, the feature combinations with small variance are discarded and those with

large variances preserved in the new feature set. If we express the occurrence of the transformed feature vector as:

$$f_r(\mathbf{y}) = f_x(x_0)^{w_0} \cdot f_x(x_1)^{w_1} \dots f_x(x_{m-1})^{w_{m-1}} f_x(x_m)^{w_m} \dots f_x(x_{n-1})^{w_{n-1}} \quad (5)$$

where x_i denotes a transformed feature and \mathbf{y} is the truncated feature vector, then the weighting scheme of figure 2(a) corresponds to:

$$w_i = \begin{cases} 1 & 0 \leq i \leq m-1 \\ 0 & m-1 < i \leq n-1 \end{cases} \quad (6)$$

It is suggested that a weighting scheme such as in (b) could prove more useful as it removes the hard decision of where to prune the feature set. This scheme would be based on the assumption that in the transient section from m_1 to m_2 that the features gradually become less useful with decreasing variance (eigenvalue). Weighting scheme (c) takes this a step further by suggesting that in the interval m_1 to m_2 that the weight of the features should be some function of their variances. One possibility is that the weights be a function of the F-Ratio of the transformed features, as the F-Ratio is a measure of separability and measures the contribution of a feature to the separability of the classes being recognised.

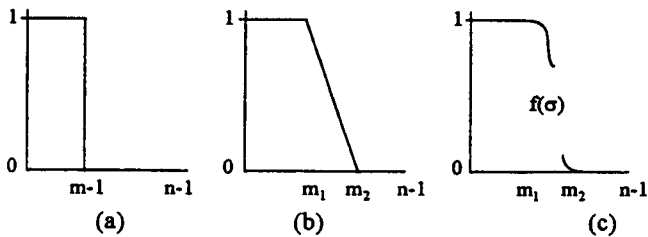


Figure 2: Subsequent Weighting schemes for Feature Set.

For the present work, the weighting scheme (a) was used to evaluate the potential of LDA for the new feature set. The transformation A was applied both training as test data and a new set of HMMs trained for the vocabulary.

5. EXPERIMENTS AND RESULTS

All experiments were carried out on the E-set (b, c, d, e, g, p, t, v) from the Connex spoken alphabet database. HMMs were trained for each letter using models with 13 emitting states, 5 mixtures per state, and 9 states tied across models. In the case of experiments on the LDA methods, models were first trained with 13 emitting states and one mixture per state with full covariance. The transform A was obtained from these models, the training and test data transformed or truncated as appropriate and new models trained and tested.

5.1 Cepstral Time matrix with Filterbank Implementation

The Average and Shift Down Sampler of the filterbank outputs every .8ms the average of the time samples within a window of

128 samples. Table 1 demonstrates the effects of increasing inclusion of the number of columns of DCT features on recognition. Results are given for matrix lengths of 25.6ms and 51.2ms, which correspond to forming a matrix by concatenation of 32 and 64 cepstral vectors respectively.

The results show that recognition results improve consistently when the longer matrix length of 51.2ms is employed. The matrix rate was the same in both cases, showing that the transitional dynamics are more useful calculated over the longer time period.

No. of columns of cepstral time matrix used.	% Recognition Matrix length 25.6ms	% Recognition Matrix length 51.2ms
Col 1	83.76	86.55
Col 1-2	87.94	89.09
Col 1-3	87.12	89.66
Col 1-4	86.55	89.25

Table 1: Use of Increasing Number of Columns of Cepstral Time Matrix.

5.2 Cepstral Time and Higher Order Dynamics.

Experiments were carried out to explore the effect of taking the long-term variation in the cepstral time matrices over successively longer segments. This was done for a value of S in equation (5) of between 9 and 25. This corresponds to taking between 5 and 12 matrices forward and backwards in time to represent the segment around a particular cepstral time vector. The experiments were carried out for original matrix lengths of 25.6ms and 51.2ms. The first column of the cepstral time matrix (12 features) formed the subsegment level features while 12 features from the first column of the resultant DCT matrix formed the segment level features.

Segment Length (S)	% Recognition Matrix length 25.6ms	% Recognition Matrix length 51.2ms
9	89.25	89.09
11	90.57	89.42
13	90.98	89.09
17	90.81	89.42
21	89.58	89.25
25	89.83	89.42

Table 2: Effect of Using Different Segment Lengths for Segmental Variation of Cepstral Time.

Referring back to Table 1 we see that the use of one column of the original cepstral time matrix, with a matrix length of 25.6ms, achieves 83.76% recognition. The additional use of the segment based dynamic features consistently improves performance by up to 7%. This greatest increase in performance was achieved when the variation was taken over 13 cepstral vectors. The increase in performance is less pronounced when the original matrix length of 51.2ms is used. On average, a 3% improvement in recognition

