

INCLUSION OF TEMPORAL INFORMATION INTO FEATURES FOR SPEECH RECOGNITION

Ben Milner

ben@saltfarm.bt.co.uk

Speech Technology Unit, BT Laboratories,
Martlesham Heath,
Suffolk, England.

ABSTRACT

Conventional methods for incorporating temporal information into speech features apply regression to a series of successive cepstral vectors to generate differential cepstra, or apply a cosine transform to generate cepstral-time matrices. This paper aims to generalise these techniques such that a series of stacked cepstral vectors is multiplied by a temporal transform matrix to produce the final speech feature. This can be made to incorporate both static and dynamic speech information. Using this method, the coding of temporal information is not restricted to regression or cosine coefficients - any suitable transform may be used. Results are presented for a variety of transforms, such as Legendre, Karhunen-Loeve, Cosine, Rectangle, where it is shown that the transform based techniques offer higher performance than conventional differential cepstrum.

1. INTRODUCTION

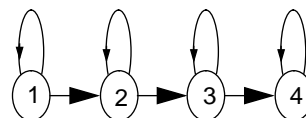
After the feature extraction stage of a speech recognition system, successive feature vectors are correlated. However a well known deficiency of HMMs is the lack of an efficient mechanism for the utilisation of this correlation. The left-right HMM provides a temporal structure for modelling the time evolution of speech spectral characteristics from one state into the next, but within each state the observation vectors are assumed to be independent and identically distributed (IID). The IID assumption states that there is no correlation between successive speech vectors. This implies that within each state the speech vectors are associated with identical probability density functions (PDFs) which have the same mean and covariance. This further implies that the spectral-time trajectory within each state is a randomly fluctuating curve with a stationary mean. However, in reality the spectral-time trajectory clearly has a definite direction as it moves from one speech event to the next. Figure 1 illustrates this restriction of HMMs. The dotted path represents the trajectory of a speech signal through the HMM feature space. As time progresses the trajectory passes smoothly through the model, exiting each state near to the boundary of the next state. The solid line shows a typical IID path through the HMM. In each state there is no temporal information, and the path randomly moves around the mean, then exits on to the next state, arriving anywhere in the next state space.

This violation of the IID assumption, by the feature vectors, contributes to a limitation in the performance of HMMs. Including some temporal information into the speech feature can lessen the effect of this assumption that speech is a stationary independent

process, and can be used to improve recognition performance.

The conventional way of including temporal information into the speech feature is to augment the cepstrum with the differential cepstrum or alternatively use cepstral-time matrices - these are reviewed in section 2. A generalisation of these techniques is presented in section 3, where an alternative method of viewing the inclusion of temporal information using a temporal transform matrix is introduced. Included in this section are some possible transforms which may be used. Section 4 presents experimental results using a selection of these temporal transforms, and a conclusion of the work is presented in section 5.

4-state HMM



HMM feature space

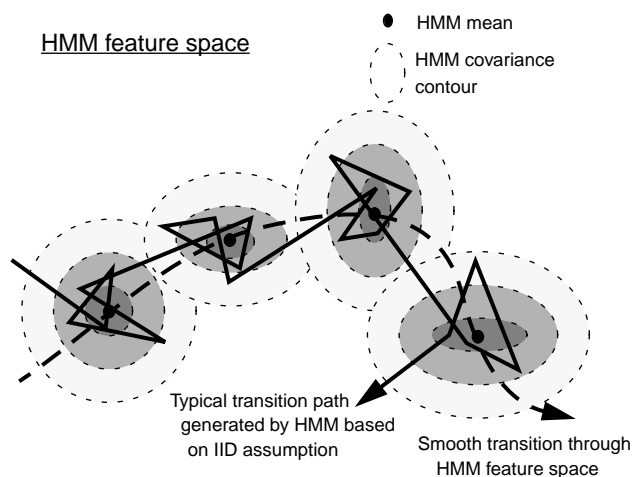


Figure 1: Transition through the HMM feature space.

2. INCLUSION OF TEMPORAL INFORMATION INTO THE SPEECH FEATURE

This section briefly reviews the differential cepstrum and cepstral-time matrices as methods of including temporal information into the feature vector.

2.1. Differential Cepstrum

The differential cepstrum is calculated by applying a weighted summation to a time sequence of cepstral vectors. The first-order cepstral derivative (velocity) is calculated using regression analysis, [1], with the general equation being,

$$\partial c_t(n) = \frac{\sum_{k=-K}^K k c_{t+k}(n)}{\sum_{k=-K}^K k^2} \quad (1)$$

where $\partial c_t(n)$ is the first time derivative of the n^{th} cepstral coefficient at time frame t , and $c_{t+k}(n)$ is the n^{th} coefficient of the $t+k^{\text{th}}$ static cepstral vector. The range $-K$ to $+K$ is the time span of cepstral vectors across which the derivative is calculated.

In a similar way, the second derivative, $\partial^2 c_t(n)$, is calculated using a weighted summation of the first derivative. This process can be continued, allowing the calculation of a derivative of any order. Hanson and Applebaum, [1], recommend a width of $K=2$ for the first derivative and a width of $K=1$ for the second derivative.

2.2. Cepstral-time matrices

A cepstral-time matrix, $C_t(m,n)$, is obtained either by applying a 2-D Discrete Cosine Transform (DCT) to a spectral-time matrix. Since a 2-D DCT can be decomposed into two 1-D DCTs, the cepstral time matrix can also be obtained by applying a 1-D DCT to a stacking of M successive MFCC speech vectors, $c_t(n)$, [2],

$$C_t(m, n) = \sum_{k=0}^{M-1} c_{t+k}(n) \cos \frac{(2k+1)m\pi}{2M} \quad (2)$$

This process is shown in Figure 2. In the cepstral-time matrix, the lower index coefficients along the quefrequency axis, n , represents the spectral envelope, whereas the higher coefficients represent the pitch and excitation, as is the case for the cepstrum. Along the pseudo-time axis, m , the lower coefficients represent the longer time variation of the cepstral coefficients, and the higher coefficients the short time variation. The column, $m=0$, represents the average or steady-state level of the spectral-time matrix.

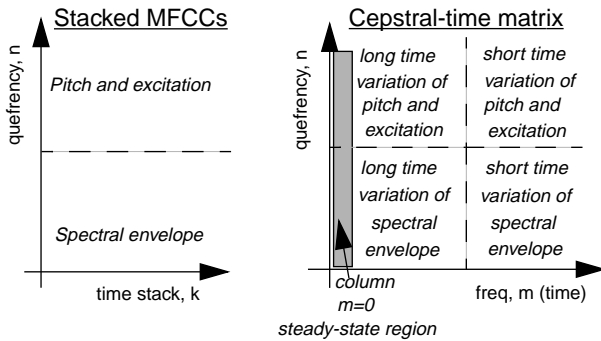


Figure 2: Regions of the cepstral-time matrix.

Typically only a sub-matrix of the cepstral-time matrix is used for recognition, thus the matrix can be truncated down $N' \times M'$. For example a useful sub-matrix is given by the first three columns, excluding the zeroth column - which contains any linear time-invariant channel distortion which may be present - [2].

3. GENERALISED TEMPORAL TRANSFORM ON STACKED VECTORS

Whilst the implementation of differential cepstrum and cepstral-time matrices seems very different, in fact they can be viewed as similar processes. Both apply a temporal transform onto a sequence of successive cepstral vectors. For example the cepstral-time matrix, \mathbf{T} , is calculated from a multiplication of a matrix containing stacked cepstral vectors, \mathbf{M} , and a matrix comprised of cosine basis functions as its columns, \mathbf{C} , i.e.

$$\mathbf{T} = \mathbf{M} \mathbf{C} \quad (3)$$

This transform can be generalised such that the cosine matrix, \mathbf{C} , is replaced by any matrix, \mathbf{H} , which can encode the temporal information, resulting in the generalised feature matrix, \mathbf{V} , i.e.

$$\mathbf{M} \mathbf{H} = \mathbf{V} \quad (4)$$

$$\begin{bmatrix} c_t(0) & c_{t+1}(0) & \dots & c_{t+M-1}(0) \\ c_t(1) & c_{t+1}(1) & \dots & c_{t+M-1}(1) \\ c_t(2) & c_{t+1}(2) & \dots & c_{t+M-1}(2) \\ \vdots & \vdots & \ddots & \vdots \\ c_t(N) & c_{t+1}(N) & \dots & c_{t+M-1}(N) \end{bmatrix} \begin{bmatrix} h_{0,0} & h_{0,1} & \dots & h_{0,M-1} \\ h_{1,0} & h_{1,1} & \dots & h_{1,M-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M-1,0} & h_{M-1,1} & \dots & h_{M-1,M-1} \end{bmatrix} = \begin{bmatrix} v_t(0,0) & v_t(0,1) & \dots & v_t(0,M-1) \\ v_t(1,0) & v_t(1,1) & \dots & v_t(1,M-1) \\ v_t(2,0) & v_t(2,1) & \dots & v_t(2,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ v_t(N,0) & v_t(N,1) & \dots & v_t(N,M-1) \end{bmatrix}$$

The columns, $\phi_i(j)$, of the temporal transform matrix, \mathbf{H} , are the basis functions for encoding the temporal information, where i indicates the column and j the element along that column. It is important that these columns are orthogonal, as this serves to decorrelate the columns of \mathbf{V} , which is particularly important when using HMMs with diagonal covariance matrices - it is assumed that the rows are already decorrelated, which is a reasonable assumption for MFCCs. To enable a dimensionality reduction, \mathbf{H} must also be selected such that useful speech information is compressed into a sub-matrix of \mathbf{V} which is used for recognition.

Using this method of encoding temporal information, a wide range of transforms can be used as the temporal transform matrix, \mathbf{H} . Indeed, the differential cepstrum, described in section 2.1 can be calculated using a temporal transform matrix. For example, to generate static, first and second order cepstral derivatives - based on equation (1) - matrix \mathbf{H} would be,

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & -2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (5)$$

with the stack width, M , set to 5. The zeroth column of the resulting feature matrix, \mathbf{V} , is the static cepstrum, the first column the first derivative, or velocity, of the cepstrum and the second column the second derivative, or acceleration.

A few examples of other polynomials are described below which can be used as the temporal transform matrix, [3].

Discrete Legendre Transform (DLT)

The Discrete Legendre Transform is determined by solving Legendre's differential equation, [4],

$$(1-x^2)y'' - 2xy' + n(n+1)y = 0 \quad (6)$$

with the resulting solution known as the Legendre polynomial of degree n , denoted by $P_n(x)$,

$$P_n(x) = \sum_{m=0}^n (-1)^m \frac{(2n-2m)!}{2^n m!(n-m)!(n-2m)!} x^{n-2m} \quad (7)$$

The basis functions of the DLT are given by $P_n(x)$, calculated for $0 \leq n \leq M-1$.

Discrete Rectangle Transform (DRT)

The Discrete Rectangle Transform is computed, in this case, by applying a three-state threshold to the DCT (+1, 0, -1), i.e.

$$r_{n,m} = \begin{cases} 1 & \text{if } c_{n,m} > 0 \\ 0 & \text{if } c_{n,m} = 0 \\ -1 & \text{if } c_{n,m} < 0 \end{cases} \quad (8)$$

where $c_{n,m}$ is a cosine transform coefficient and $r_{n,m}$ is the corresponding rectangle transform coefficient. The advantage of the DRT is that it is very fast to compute as it can be implemented using only additions and subtractions.

Karhunen-Loeve Transform (KLT)

The previous temporal transforms have all been based on pre-determined polynomials. These transforms, however, will not necessarily optimally encode the temporal information, i.e. diagonalise the covariance matrix of \mathbf{M} . The optimal transform for encoding temporal information is the Karhunen-Loeve Transform, [3]. To compute the KLT, statistics regarding the temporal correlation of the stacked cepstral vectors in \mathbf{M} must be determined. To do this M -dimensional vectors, \mathbf{x}_t , composed of successive cepstral coefficients, are created from the stacked cepstral vectors,

$$\mathbf{x}_t = [c_t(n), c_{t+1}(n), \dots, c_{t+M-1}(n)] \quad (9)$$

By pooling together many examples from the speech training data

the covariance, $\Sigma_{\mathbf{x}_t}$, of the stacked cepstral coefficients can be determined. The KLT temporal transform matrix, \mathbf{H} , is given from the eigenvectors of the covariance matrix,

$$\mathbf{H}^T \Sigma_{\mathbf{x}_t} \mathbf{H} = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_{M-1}) = \Delta \quad (10)$$

The eigenvectors which form the KLT basis functions are ranked in terms of their eigenvalues. The zeroth KLT basis function corresponds to the eigenvector which has the largest eigenvalue.

Figure 3 shows the first 5 basis functions of each of the temporal transforms described (DCT, DLT, DRT and KLT). The most obvious conclusion from these is that they all have very similar characteristics. In particular it was noted that the DCT and KLT (calculated across approximately 70000 frames of speech) were almost identical - this was also noted in [5] for a speech coding application.

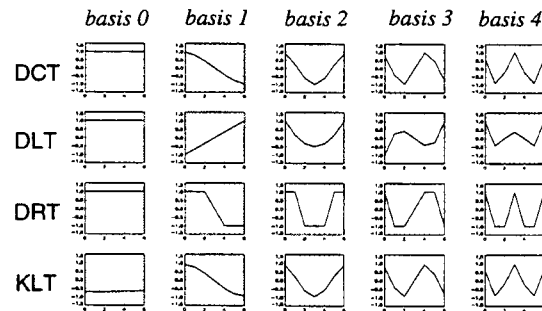


Figure 3: Basis functions of the DCT, DLT, DRT and KLT temporal transforms.

4. EXPERIMENTAL RESULTS

This section describes a range of experiments based on the generalised temporal transform of equation (4). In each experiment the stacked cepstral matrix, \mathbf{M} , is formed from a stack of 7 9-D MFCCs (coefficients 0 to 8). The experiments are conducted on a speaker independent isolated digit task using a 1000 talker telephony speech database. The digit vocabulary consists of the digits *one, two, ..., nine, oh, zero* and *nought* and is modelled using 6-state, 7-mode diagonal covariance HMMs.

As a baseline experiment, the identity matrix is used as the temporal transform matrix. This has the effect that the resultant speech feature consists of stacked cepstral vectors and hence contains no implicitly encoded temporal information.

Several different polynomials are used as the temporal transform matrix. These include the discrete cosine transform (DCT), which produces cepstral-time matrices, the discrete Legendre transform (DLT) and the discrete rectangle transform (DRT). In addition data-derived transforms are also included such as the Karhunen-Loeve transform (KLT) and linear discriminant analysis (LDA). All these use columns 1 to 3 of matrix \mathbf{V} as the speech feature, which are effectively the first, second and third time derivatives -

resulting in a 27 dimensional speech feature.

LDA, [6], is applied to the feature comprised of 7 stacked cepstral vectors, matrix \mathbf{M} . The LDA derived speech feature is produced by multiplying the 7x9-D stacked MFCCs by the LDA transform matrix, and subsequently truncating the feature down to 27 dimensions. Using LDA to encode the matrix of stacked cepstral vectors, \mathbf{M} , results in a $NM \times NM$ transform matrix which is much larger than $M \times M$ temporal transform matrix of equation (4). As a result LDA does not fit exactly into the temporal transform framework, but is included as it offers a method of encoding the information contained within \mathbf{M} , based on discrimination.

For comparison a few conventional differential cepstral features are included in the results.

The results are detailed in Table 1, where the \oplus operator is used to represent the augmentation of vectors.

Stack transform, \mathbf{H}	Feature size	Recognition accuracy
Identity, 3x3	27	83.7 %
Identity, 5x5	45	87.6 %
Identity, 7x7	63	87.7 %
Cosine, DCT (CTM)	27	96.9 %
Legendre, DLT	27	97.1 %
Rectangle, DRT	27	96.7 %
Karhunen-Loeve, KLT	27	97.4 %
LDA	27	95.7 %
c (MFCCs 0 to 8)	9	79.0 %
$c \oplus \partial c$	18	92.5 %
$c \oplus \partial c \oplus \partial \partial c$	27	93.4 %
$c \oplus \partial c \oplus \partial \partial c \oplus \partial \partial \partial c$	36	93.5 %
∂c	9	92.5 %

Table 1: Recognition performance using a selection of temporal transforms.

The most obvious conclusion from the results is that including temporal information into the speech feature improves recognition performance. The most simple feature comprises only the static cepstrum (c in the table), and attains worst performance - 79.0 %. Including the neighbouring static cepstrum with this (i.e. using the identity matrix as the temporal transform matrix) improves performance. Using 3 successive cepstral vectors improves performance to 83.7 % and with 7 successive vectors the performance reaches 87.7 %. These are both significantly higher than with a single cepstral vector, as there is now temporal information in the speech feature although as yet not encoded efficiently.

By applying one of the temporal transforms (DCT, DLT, DRT, KLT) to the stacked cepstral vectors improves performance considerably - to over 96 % - as well as enabling a reduction in the feature size. Best performance of 97.4 % is given by the KLT which is unsurprising given that the KLT is the theoretical optimal transform. LDA improves performance but not by as much as the previous transforms. In other experiments LDA has been applied to cepstral-time matrices themselves. Whilst not improving performance, it has allowed a significant reduction in feature size - over 97.1 % with a 15-D feature, and 94.4 % with a 5-D feature.

Considering now conventional cepstral features it is shown that augmenting the static cepstrum with differential cepstra also improves performance - adding velocity cepstra ($c \oplus \partial c$) increases performance from 79.0 % to 92.5 %. Adding higher order derivatives again improves performance, but this tends to fall off at the 3rd derivative - as well as increasing the feature size. It is interesting to note that using just velocity cepstra (∂c) gives 13.5 % higher performance than just the static cepstra (c). These can be computed either conventionally using regression, equation (1), or in the temporal transform framework, equation (4).

5. CONCLUSION

The aim of this paper has been to generalise the incorporation of temporal information into the speech feature by viewing it as the application of a temporal matrix transform on a matrix comprised of stacked cepstral vectors. As a result of this a wide range of transforms have been demonstrated which can be used to encode the temporal dynamics of speech.

Results have shown that increased performance is attainable when temporal information is encoded by one of the implicit transforms, such as DCT, DLT, KLT, compared to using the conventional differential cepstra. Best performance is achieved by the KLT, which has basis functions almost identical to those of the DCT, making cepstral-time matrices a good approximation of idealised temporal features.

6. REFERENCES

1. B.A. Hanson and T.H. Applebaum, "Robust speaker-independent word recognition, using static, dynamic and acceleration features", Proc. ICASSP, pp. 857-860, 1990.
2. B.P. Milner and S.V. Vaseghi, "An analysis of cepstral-time feature matrices for noise and channel robust speech recognition", Proc. Eurospeech, pp. 519-522, 1995.
3. N.S. Jayant and P. Noll, Digital coding of waveforms, Prentice-Hall, 1984.
4. E. Kreyszig, Advanced Engineering Mathematics, John Wiley and Sons, 1983.
5. N. Ahmed, "Discrete cosine transform", IEEE Trans. Computing, January 1974.
6. E.S. Parris and M.J. Carey, "Estimating linear discriminant parameters for continuous density hidden markov models", Proc. ICSLP, pp. 215-218, 1994.