A SPEAKER INDEPENDENT PHONEME RECOGNITION SYSTEM

Andrew Tridgell and Bruce Millar

Computer Sciences Laboratory
Research School of Physical Sciences and Engineering
Australian National University

ABSTRACT - A speaker independent phoneme recognition system is presented and discussed. Some of the unique features of this system include the use of a tree based vector quantiser and the use of multiple vector quantisers for each parameter set.

INTRODUCTION

This paper describes a speech recognition system that has been built in the Computer Sciences Laboratory of the Australian National University. The system is a speaker independent phoneme recognition system for continuous speech based around a multiple layer discrete HMM (Hidden Markov Model.)

Although the basis of the system is similar to several other published systems there are several unique features of this system which warrant discussion. In this paper an overall description of the system will first be given, followed by a brief discussion of some of its novel features. It will be assumed that the reader has at least a passing familiarity with discrete HMMs applied to speech recognition. For a good description of these techniques see Rabiner (1989).

The system has been evaluated on the TIMIT speech corpus. The system is trained on 1000 sentences randomly chosen from a set of 3696 sentences. These sentences are spoken by 462 speakers from 8 dialect regions of American English. Testing is carried out on 100 sentences randomly chosen from 1344 sentences spoken by 168 speakers, with no overlap between training and testing speakers.

OUTLINE OF THE SYSTEM

The best way to describe the system is to follow an example spoken utterance through the stages of processing until the required phonetic transcription is produced. For this purpose an already trained model will be assumed, and only brief mention will be given to the techniques used to derive the parameters of the model.

The input to the system is a sampled speech utterance with a sample rate of 16,000 samples per second and a resolution of 16 bits. These samples are divided into overlapping frames of 320 samples each, with a frame advance of 80 samples.

The first stage involves the application of several pre-processing techniques to the frames of sampled speech. In a typical configuration a total of five different pre-processing techniques are applied to each of the speech frames, yielding five parameter vectors for each frame. These five techniques, in no particular order, are:

- Pre-emphasise with a constant of 0.97, apply a Hamming window, find 20 auto-regression coefficients, and finally transform these coefficients to 20 cepstral coefficients.
- Take a FFT of the sample frame padded to 512 samples and divide the resulting magnitude array into 32 bins of equal size covering only the bottom half of the spectrum. The sum of squares of the elements in each of these bins is then found and the log of these coefficients is taken. Each frame is then represented by 32 log-band-power estimates.
- The difference between the values of the cepstral coefficients for the frame 2 past the current frame and the cepstral coefficients of 2 frames before the current frame is found yielding 20 delta-cepstral coefficients.

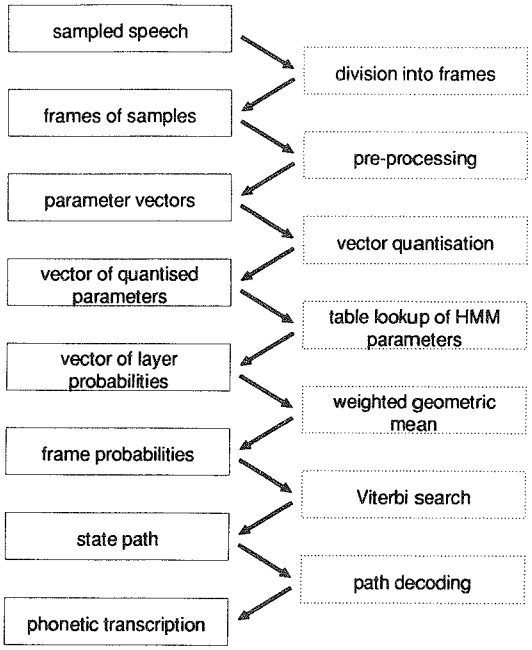

Figure 1. The processing of speech data

- An identical differencing procedure is applied to the band-power estimates to give 32 delta-band-power values.
- The zero-crossing rate and log frame energy are found and concatenated to give a final 2 element parameter vector.

The 5 parameter vectors obtained from the previous stage must then be vector quantised to produce a vector of integer values. Between 1 and 5 vector quantisations are obtained for each of the parameter vectors yielding a quantised vector of K integer values (typically 15). The application of more than one vector quantisation to each parameter vector is unusual but has been found to be very effective in reducing the effect of quantisation error on the performance of the system. This is discussed in more depth in a later section of this paper.

The vector quantiser that is used is a median split linear discriminant analysis tree. This vector quantiser has several desirable properties that make it particularly suitable for this system. The precise algorithm and some of these properties are discussed in a later section of this paper. Each quantiser produces an integer in the range $[0,2^M)$ where M is typically 9. Due to one of the properties of the quantiser each of these values has equal probability, so the quantiser divides the parameter spaces into 512 regions of equal probability.

It is at this stage that the parameters of the core of the HMM start to be used. Each state within the HMM has associated with it a table giving the probability of expecting each of the 512 possible quantised values

for each of the quantised parameter sets at that specific state of that specific phone model. It is these tables that make the HMM so memory intensive. With typically 9 states per phone model and 39 phone models these tables take nearly 11Mb of memory. Each entry of the tables consists of a log integer value to facilitate the fast calculation of products of these values.

These tables are indexed by the quantised input parameters yielding a layer by layer probability value for the frame. After these tables have been applied each state of the HMM will have associated with it a vector of probabilities of length K stored as log integers. A weighted arithmetic mean of these log values is then taken to give something akin to the log of the geometric mean of the probabilities. Currently only manually set or equal weights are used in computing the arithmetic mean, with the problem remaining as how to automatically estimate optimal values. This problem is discussed in a later section of this paper.

At this stage each of the HMM states has a frame probability for each of the frames in the speech sample indicating the probability that that specific frame is a part of that specific state. Also associated with each state are the state transition lists and probabilities. These specify what states each state can make a transition to, and what the probability of that transition is. These two sets of probabilities can then be combined using the Viterbi algorithm to yield the path through the states that has the highest probability of occurrence. This state path is a sequence of state numbers (unique state identifiers) that best match the input sequence of speech frames.

This concludes the core of the HMM routines, with the only task remaining to decode the state path into a phone path and produce the phonetic transcription of the utterance. The decoding is a simple matter of replacing the state number of each element in the path by the phone label corresponding to the phoneme of which that state is a part. Consecutive occurrences of the same phone label can then be removed to give the phone path. This can then be compared with the labeling provided in the TIMIT database using a dynamic programming match to determine the accuracy of the recognition.

All of the above processing from sample speech data to phonetic labels can be achieved with the current C implementation on a SparcStation 2 in approximately 1/3 real time. This means that with a higher end workstation real time performance could be reasonably expected.

VECTOR QUANTISATION USING LINEAR COMBINATION TREES

In a recent paper the authors described a technique whereby a decision tree can be used to replace the unsupervised clustering algorithms commonly used for vector quantisation (Tridgell & Millar, 1992). It was demonstrated in particular that a binary decision tree grown using the single variable split methods used in the CART (Breiman et al, 1984) system provided a discrete HMM front end competitive with that produced with a sophisticated gaussian mixture model (Zhang & deSilva, 1991).

Since that paper a tree growing method based upon linear combination splits has been developed which has several advantages over the single variable splits. Results have shown that this new technique provides substantially superior performance over the single variable split trees.

The new method is quite simple in both concept and implementation, with the linear combination coefficients coming from Linear Discriminant Analysis (Cooley & Lohnes, 1971) rather than multi-dimensional optimisations. This means that the resulting trees are deterministic and that they can be produced quickly. Experiments with alternate "optimising" algorithms have thus far yielded disappointing results when applied to discrete HMMs.

The tree is built by recursively bisecting the data into regions of equal probability. The hyper-plane that divides a body of data is defined by the first linear discriminant function for that data and the median of the values taken by the first linear discriminant function on the data. If this procedure is then applied N times then the input space will be divided into 2^N equally probable regions.

Once the tree has been built each of the resulting regions can be arbitrarily assigned an integer in the range $1...2^N$ These assignments correspond to the output of a conventional vector quantiser. When a new set of data is then presented to the tree, each sample vector can be assigned an integer corresponding to the region of the input space in which it lies. This evaluation stage requires N inside

product and comparison calculations to be performed.

Recursive partitioning of data using linear combination split trees is not new in the statistics community. The CART system mentioned previously has a linear combination split capability, as does the FACT system (Loh & Vanichsetakul, 1988). These systems employ complex multi-dimensional optimisations that attempt to maximise a measure of the purity of tree leaves, while minimising the tree complexity. It is the computational complexity of these systems that precludes then from being used as a vector quantiser for a HMM phoneme recogniser. After many days of CPU time on a fast workstation a tree produced by the CART linear combination system performed considerably worse than one produced in minutes using the median LDA splits described above.

For some applications it can be required that the tree building algorithm be unstable so that multiple trees can be built using one data set. If this is the case then a certain amount of randomness can be easily built into the tree building procedure. This is done by randomly choosing between the first and second linear discriminant function as each split is made. For a tree of depth N this leads to $2^{2^{(N+1)}-1}$ possible trees from one data set. Experiments have shown that these "secondary" trees perform very similarly to the "primary" tree produced only using the first linear discriminant function. One application of an unstable tree building algorithm is given below.

MANY-LAYER HIDDEN MARKOV MODELS

For several years multiple layer (multiple codebook) Hidden Markov Models have been very successful for speech recognition tasks (Huang, Hon & Lee, 1989). In these systems the frame probability required by the model is replaced by a combined probability estimate from several sources. These sources will typically take the form of several different pre-processing techniques applied to the raw speech frames, providing several sets of parameters.

In the case of a discrete Hidden Markov Model these parameter sets are quantised independently by different vector quantisers to produce a set of integer values, one for each layer in the model. These integer values are then converted to probability estimates using a table that has been constructed during the models training phase. Finally these probabilities are combined to form a frame probability estimate

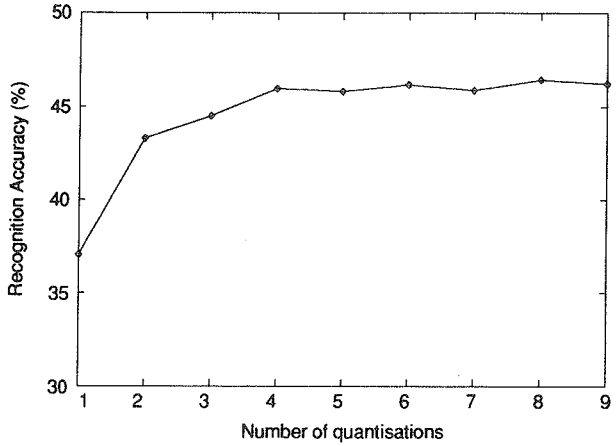

Figure 2. The effect of adding vector quantisations for a single parameter set.

using a geometric or arithmetic mean.

Lee (1988) demonstrated that the use of multiple layers in this way could increase the resolution of the quantisation. This is because 2 layers of size N each can produce N^2 possible probabilities, whereas a single combined layer can take on only 2N values. The amount of training data required to accurately estimate these probabilities will, however, only be that for a single layer of size N.

Using discrete Hidden Markov Model phoneme recognition as an example we have been able to demonstrate that this observation can be taken a lot further. In particular we have been able to show that better performance can be obtained by the use of several vector quantisers for each set of pre-processed parameters. This uses the fact that many vector quantisation algorithms are inherently unstable, so that marginally different data sets can produce quite different partitionings of the vector space.

In figure 2 we show the recognition accuracy for 39 phonemes as the number of quantisations for a single parameter set is increased. In this case we are using 20 cepstral coefficients. The results are for the TIMIT speaker-independent continuous speech database. The percentage accuracy refers to the number of phones correctly recognised minus the number of insertions divided by the total number of phones. When the complete set of input parameters is used results of over 60% have been achieved.

In this example we are combining the probability estimates from individual layers by taking the geometric mean of the individual probabilities. For more complex combinations where differing numbers of layers are used for each parameter set it is not at all clear that this method is optimal. In particular the geometric mean assumes that each probability estimate should be given equal weight. This can be expressed as:

$$\log P_{frame} = \frac{\sum W_L \log P_L}{\sum W_L}$$

if the W_L take on equal values. In some cases it seems clear that an optimal set of layer weights, W_L, may not give equal weight to all layers. There seems, for example, no reason why delta-Cepstral coefficients should have the same weight as Cepstral coefficients. With this in mind a method was sought to try and optimise the values of the layer weights.

The only success to date in this regard has been by manually setting the values of the weights on a phone by phone basis. This gave increased recognition accuracy but has thus far shed no light on methods for systematically optimising the weight values. It is thought that if such a method could be found that it could be of considerable benefit.

WORK IN PROGRESS

Work is currently proceeding to try to incorporate other recognition systems as inputs to the system. In particular it is hoped that the large recurrent neural networks developed by Robinson (1991) will provide a valuable addition to the recognition accuracy of the system. This continues the theme of the project, which is to use Hidden Markov Models as a firm statistical framework for the combination of results from many sources.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Australian Telecommunications and Electronics Research Board, ANUTech Pty Ltd and the Commonwealth of Australia who provided postgraduate scholarships to the first author.

REFERENCES

Breiman L., Friedman J.H., Olshen R.A. & Stone C.J. (1984) *Classification and Regression Trees*, (Wadsworth International Group: California)

Cooley W.W. & Lohnes P.R. (1971) *Multivariate Data Analysis*, (John Wiley and Sons: New York)

Huang X.D., Hon H.W. & Lee K.F. (1989) *Multiple Codebook Semi-Continuous Hidden Markov Models for Speaker-Independent Continuous Speech Recognition*, Carnegie Mellon University Tech. Rep., CMU-CS-89-136

Lee K.F. (1988), *Large-vocabulary speaker-independent continuous speech recognition: The SPHINX system*, PhD thesis, Dept. of C. Science, Carnegie Mellon University

Loh W.Y & Vanichsetakul N. (1988) *Tree structured classification via generalized discriminant analysis*, Journal of the American Statistical Association, Vol. 83, No. 402 pp715-728

Rabiner, L.R (1989) *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proc. IEEE, Vol. 77, No. 2, pp257-286

Robinson, T & Fallside, F (1991) *A Recurrent Error Propagation Network Speech Recognition System*, Computer Speech and Language, Vol. 5 No. 3

Tridgell, A.J & Millar, J.B (1992) *Alternative Pre-Processing Techniques for Discrete Hidden Markov Model Phoneme Recognition*, Proc. ICSLP-92 [forthcoming]

Zhang Y. & deSilva C.J.S. (1991) *An isolated word recogniser using the EM algorithm for vector quantisation*, Proc. IREECON 91, Sydney