



## PRODUCT CODE VECTOR QUANTISATION AND HIDDEN MARKOV MODELLING IN ISOLATED WORD RECOGNITION

Børge Lindberg\*, Paul Dalsgaard\*

### ABSTRACT

This paper presents a speaker independent isolated word recogniser, which combines the product codebook vector quantisation principle with the discrete hidden Markov modelling (HMM), so that each frame in the unknown test word (or training word) is described by two symbols, the linear predictive coding (LPC) shape and gain. The recogniser (both training and testing) has been evaluated on a 12 word vocabulary. The recognition results as well as the implementation requirements are discussed and compared with other approaches to speaker independent isolated word recognition.

### INTRODUCTION

The theory of hidden Markov models has in the last decade successfully been applied in the area of speech recognition, i.e. isolated word recognition, connected word recognition and continuous speech recognition. The reason for this is the ability of HMM's to characterise the inherent nonstationarity of speech signals. In each state of the HMM, a probabilistic function is defined in order to model the statistics of the observed symbols in the training material. Initial results (ref 1) using discrete probability distributions showed an excellent ability to perform speaker independent isolated word recognition, and these results were later improved by using continuous Gaussian density functions (ref 5) and continuous Gaussian mixture density functions (ref 6,7).

In the discrete HMM, the observed symbols are normally drawn from an alphabet,  $\mathbf{V}$ , of  $m$  prototypical spectra,  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m$ , by vector quantising the original short time speech power spectra. It is often stated that the advantage of continuous HMM's over discrete HMM's is due to the inherent quantisation error, which occurs when trying to represent the continuous speech spectra by the vectors from the alphabet. This quantisation error could in theory be eliminated by extending the number of vectors in the codebook, but due to the limited amount of training data, this would imply poorer estimation of the discrete probability distributions (often referred to as symbol probabilities). In this paper it will be investigated, whether the product codebook, which combines the LPC shape and gain, has the ability to minimise the quantisation error and still provide sufficient training data for reestimating the symbol probabilities in the HMM.

### PRODUCT CODE VECTOR QUANTISATION

The type of product codebook which will be considered is the so called shape-gain codebook. It has been shown (ref 2) that adding gain information to the LPC shape information can give improvements in recognition accuracy over that obtained using the LPC shape information alone. A shape-gain codebook is characterised by a shape codebook  $\mathbf{S}$  and a gain codebook  $\mathbf{g}$ . The shape codebook:  $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{N_S}\}$  contains  $N_S$  prototype spectra, which encode the spectral information. The vector  $\mathbf{s}_i$  represents a gain normalised AR-model of prediction order  $p$ :

$$\mathbf{s}_i \leftrightarrow 1/S_i(z), \quad S_i(z) = 1 + \sum_{j=1}^p s_{ij} z^j, \quad 1 \leq i \leq N_S \quad (1)$$

The gain codebook:  $\mathbf{g} = \{g_1, g_2, \dots, g_{N_g}\}$  contains  $N_g$  prototype gain values, which encode the gain information. Note that by means of  $(N_S \times N_g)$  elements the shape-gain codebook  $\mathbf{S} \times \mathbf{g}$  contains  $(N_S N_g)$  elements and that it is possible separately to specify the amount of quantisation of the input shape and gain.

Quantisation by means of a shape-gain codebook is the process of finding the nearest neighbour to the

\* Speech Technology Centre, Institute of Electronic Systems, Strandvejen 19, DK-9000 Aalborg, Denmark

short-time speech power spectrum  $|X(e^{j\omega})|^2$ , represented by its LPC model  $G_X(z)$ :

$$G_X(z) = \sqrt{\alpha_X}/A_X(z), \quad A_X(z) = 1 + \sum_{j=1}^P a_{Xj} z^{-j} \quad (2)$$

The nearest neighbour in the shape-gain codebook is defined as the pair  $(s_i, g_j)$  which minimises the Itakura-Saito distortion (ref 3):

$$d_{IS}(|G_X|^2, (g_j^2/|S_i(z)|^2)) = (\alpha_i/g_j^2) - \ln(\alpha_X/g_j^2) - 1 \quad (3)$$

$$\alpha_i = r_X(0) r_{S_i}(0) + 2 \sum_{k=1}^P r_X(k) r_{S_i}(k) \quad (4)$$

$r_X$  and  $r_{S_i}$  are the autocorrelations of the speech signal and the shape codebook vector  $S_i$  respectively. It can be shown (ref 3) that this nearest neighbour finding can be viewed as a two-step procedure: First an optimal shape vector  $S_i$  is found from (4), and then, based on  $\alpha_i$ , an optimal  $g_j$  is found from (3).

Using the distortion measure of (3) one can devise an algorithm for choosing a set of  $N_S$  shape vectors, and a set of  $N_g$  gain values which in combination minimise the overall distortion, when quantising the training vectors by the shape-gain codebook. The principle of the algorithm is to separately design the shape and the gain codebook by first designing the shape codebook based on the set of training vectors, and then designing the gain codebook based on the residual energies (determined in (4)), which come from quantising the training vectors by the designed shape codebook. The design algorithms in both the shape and the gain codebook case are performed iteratively. This is described in (ref 4): starting with an initial guess of  $N$  codebook elements, each vector (or residual energy) from the training material is assigned to the closest element. The centroids of the  $N$  clusters obtained in this manner are then used as new trial elements. This iteration process is continued until the overall distortion of the codebook does not change significantly. Following this the number of elements,  $N$ , in the codebook is extended by splitting all of the existing codebook elements and the iteration procedure is repeated. The splitting procedure continues until the desired codebook size is reached.

#### THE DISCRETE SHAPE-GAIN HMM

A Markov process resides at time  $t$  in state  $q_t$ , which can be one of the  $N$  states:  $Q = \{Q_1, Q_2, \dots, Q_N\}$ , and is generally described by the state transition matrix  $A = \{a_{ij}\}$ , and by the initial state distribution vector  $P = \{p_i\}$ :

$$\begin{aligned} a_{ij} &= \text{Prob}[q_{t+1} = Q_j | q_t = Q_i], & 1 \leq i, j \leq N \\ p_i &= \text{Prob}[q_1 = Q_i], & 1 \leq i \leq N. \end{aligned} \quad (5)$$

The type of Markov process which will be considered, is the so called left-to-right, which is characterised by

$$a_{ij} = 0, \quad j < i \quad \text{and} \quad p_i = \begin{cases} 0 & i \neq 1 \\ 1 & i = 1. \end{cases} \quad (7)$$

When using both the gain and the shape indices as observation symbols, it must be decided how to attach the symbols to the hidden Markov chain. Clearly a strong correlation exists between the gain and the shape information. In the modelling approach we have chosen, the dependency between the gain and the shape symbols is controlled by the underlying Markov chain, such that in each state of the Markov process two symbols are generated independently: a gain symbol and a shape symbol. In general this approach could be extended to include multiple symbols being generated in each state - each symbol describing any significant feature of the speech signal. Thus the discrete shape-gain HMM is further described by the shape symbol probability distribution in state  $j$ ,  $B = \{b_j(k)\}$ , and the gain symbol probability distribution in state  $j$ ,  $C = \{c_j(m)\}$ :

$$b_j(k) = \text{Prob}[s_k \text{ at } t | q_t = Q_j], \quad 1 \leq j \leq N, \quad 1 \leq k \leq N_S \quad (8)$$

$$c_j(m) = \text{Prob}[g_m \text{ at } t \mid q_t = Q_j], \quad 1 \leq j \leq N, 1 \leq m \leq N_g \quad (9)$$

As mentioned above gain and shape symbols are considered to be generated independently in each state. Thus

$$\text{Prob}[(s_k, g_m) \text{ at } t \mid q_t = Q_j] = b_j(k) c_j(m), \quad 1 \leq j \leq N, 1 \leq m \leq N_g, 1 \leq k \leq N_S \quad (10)$$

With the specified dependency, only minor changes have to be done in the algorithms, which are normally used for training and recognition with the discrete single symbol HMM. Based on an observation sequence  $\mathbf{OS} = \{OS_1, OS_2, \dots, OS_T\}$  of shape symbols and an observation sequence  $\mathbf{Og} = \{Og_1, Og_2, \dots, Og_T\}$  of gain symbols, both of length  $T$ , the Viterbi algorithm (ref 1) is used in the recognition process to evaluate the likelihood-score for each word in the vocabulary. The evaluation is based on an initialisation (11) followed by a recursion (12):

$$\varphi_1(1) = \ln[b_1(OS_1)] + \ln[c_1(Og_1)] \quad (11)$$

$$\varphi_1(i) = -\infty, \quad 1 < i \leq N$$

$$\varphi_t(j) = \max_{j-N_t \leq i \leq j} \{\varphi_{t-1}(i) + \ln a_{ij}\} + \ln[b_j(OS_t)] + \ln[c_j(Og_t)], \quad 2 \leq t \leq T, 1 \leq j \leq N. \quad (12)$$

The individual numbers of legal transitions in each state in the left-to-right Markov chain are assumed to be equivalent and denoted  $N_t$ . The score is evaluated for each word in the vocabulary, and the vocabulary word, which gives the highest  $\varphi_T(N)$ , is assigned to the unknown test word. Similarly the Baum Welch reestimation algorithm (ref 1) for training the HMM's can be changed.

#### IMPLEMENTATION REQUIREMENTS

Defining implementation requirements as the sum of memory and computations required for recognition (not training), the following recognisers will be compared 1) discrete single symbol HMM (DHMM), 2) discrete shape-gain HMM (DSGHMM), 3) continuous HMM (CHMM) and 4) Dynamic Time Warp (DTW) (ref 6,8). The type of CHMM, which will be considered is the Gaussian mixture density HMM, i.e. the density function for observing the speech signal representation vector  $\mathbf{x}$  (e.g. cepstrum) in state  $j$  is defined by:

$$b_j(\mathbf{x}) = \sum_{k=1}^M c_{jk} \Gamma(\mathbf{x}, \mathbf{u}_{jk}, \mathbf{C}_{jk}). \quad (13)$$

$\Gamma(\mathbf{x}, \mathbf{u}, \mathbf{C})$  denotes a  $D$ -dimensional normal density function of mean vector  $\mathbf{u}$  and covariance matrix  $\mathbf{C}$  and  $M$  denotes the number of mixture densities, which are scaled with the mixture gains  $\{c_{jk}\}$ . The type of DHMM which will be considered is the one which uses only the shape information, i.e. the corresponding codebook consists of  $N_S$  gain normalised prototype spectra, and the quantisation is done by the nearest neighbour finding specified in (4). Denoting  $L$  as the number of words in the vocabulary,  $T$  as the average number of frames in the test words and  $T_W$  as the number of templates per word in the DTW-recogniser the following memory and computation requirements can be shown approximately to hold (with typical values of  $L=12$ ,  $T_W=12$ ,  $T=40$ ,  $D=9$ ,  $N=5$ ,  $N_t=3$ ,  $M=1$ ,  $N_S=64$  and  $N_g=16$  the corresponding requirements are shown in parantheses):

Type	Memory		Number of computations	
DHMM	$NL(N_S+N_t-1)+DN_S$	(= 4,536)	$NTL(N_t+1)+TDN_S$	(= 32,640)
DSGHMM	$NL(N_S+N_G+N_t-1)+DN_S+N_G$	(= 5,512)	$NTL(N_t+2)+TDN_S$	(= 35,040)
CHMM	$NL(MD^2+MD+N_t-1)$	(= 5,520)	$NTL(N_t+MD^2+MD)$	(= 223,200)
DTW	$T_W TDL$	(= 51,840)	$DLT_W T^2/3$	(= 691,200)

It can be concluded, that when the number of words in the vocabulary is increased, it is more advantageous (in respect of the number of computations) to use the discrete HMM's instead of the continuous HMM or the DTW recogniser. It is moreover seen that there is only a minor difference in the implementation requirements between the two discrete HMM methods, which means that including gain information in a discrete HMM recogniser by means of a product codebook does not essentially affect the implementation requirements.

## RESULTS

The database on which the discrete shape-gain HMM technique could be assessed was limited to the amount of 12 words (the danish words "0", "...", "9", "start" and "stop"), each spoken by 20 female and 40 male in a sound proof booth. Thus by means of 720 tokens it is very difficult to achieve statistically significant results. The basic signal processing was done by 8 kHz sampling followed by an 8' order LPC-analysis of overlapping sections of 45 ms every 15 ms. The recognisers DHMM, DSGHMM and CHMM described in the previous paragraph have been evaluated on three different subsets of the database (20, 40 and all the 60 speakers).

Training material	DSGHMM $N_S=64, N_G=16$	DSGHMM $N_S=128, N_G=16$	DHMM $N_S=64$	DHMM $N_S=128$	CHMM $M=1, \text{ full cov.}$
20 speakers	30 (1)	40 (3)	45 (4)	44 (4)	38 (0)
40 speakers	17 (7)	16 (10)	28 (13)	16 (10)	9 (0)
60 speakers	-- (9)	-- (7)	-- (20)	-- (13)	-- (1)

*Table 1 Number of recognition errors, when 3 different types of recognisers are tested against speakers outside the training material. In parantheses is shown the results from tests on the training material.*

From Table 1, which shows the recognition results, it can be concluded that the best recogniser performance is achieved by the CHMM. It is strongly indicated that using a product codebook of 64 shape vectors and 16 gain values gives better performance than that obtained by a conventional DHMM recogniser using 128 shape vectors. There is only a slight improvement noticed when increasing the number of shape vectors from 64 to 128 - in fact a poorer performance was achieved when testing on 20 and 40 speakers. This effect must be due to insufficient amount of training data in the 128 shape vector case.

In our future work with the use of discrete shape-gain HMM's the recogniser will be assessed on a larger database (of 50 words, 120 speakers), providing more statistically significant results. Further refinements of the method, which is needed if it has to be comparable with the performance of the CHMM, will be concentrated on improving the reestimation of the symbol probabilities by means of smoothing techniques.

## REFERENCES

1. L.R. Rabiner, S.E. Levinson, M.M. Sondhi: "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent Isolated Word Recognition", Bell System Tech. Jour., April 1983
2. L.R. Rabiner, M.M. Sondhi, S.E. Levinson: "A Vector Quantizer Incorporating Both LPC Shape and Energy", Proceedings ICAASP 1984, paper 17.1
3. Michael J. Sabin, Robert M. Gray: "Product Code Vector Quantizers for Waveform and Voice Coding", IEEE Trans. on ASSP, Vol. ASSP-32, No. 3, June 1984
4. Yoseph Linde, Andres Buzo, Robert M. Gray: "An Algorithm for Vector Quantizer Design", IEEE Trans. on Communication, Vol. COM-28, No. 1, January 1980
5. Alan B. Poritz: "Linear Predictive Hidden Markov Models And The Speech Signal", Proc. ICASSP 82, pp. 1291-1294, May 1982.
6. L.R. Rabiner, B.H. Juang, S.E. Levinson, M.M. Sondhi: "Recognition Of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities", Bell System Tech. Jour. No. 6, July-August 1985
7. B.H. Juang, L.R. Rabiner: "Mixture Autoregressive Hidden Markov Models for Speech Signals", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-33, No. 6, Dec. 1985
8. J.G. Wilpon, L.R. Rabiner: "A modified K-Means Clustering Algorithm for Use in Speaker Independent Isolated Word Recognition", IEEE Trans. on ASSP, Vol. ASSP-33, No. 3, June 1985