Proceedings of The Institute of Acoustics

EXPERIMENTS IN SPEAKER-DEPENDENT ISOLATED DIGIT RECOGNITION USING HIDDEN MARKOV MODELS

Martin J Russell and Anneliese E Cook

Speech Research Unit, RSRE, St Andrews Rd, Malvern, Worcs. WR14 3PS

INTRODUCTION

Currently the most computationally useful speech pattern modelling paradigm for automatic speech recognition is based on *hidden Markov models* (HMMs). Commercial recognisers which exploit these techniques already exist and have been shown to outperform recognisers based on more traditional template matching methods [1], [2]. There is little doubt that this trend will continue and that a significant number of commercial systems in the near future will be based on HMM technology.

This success can be attributed to several factors. HMMs are a statistical formalism for modelling time-varying sequences which evolve through a set of quasi-stationary states of varying duration. As such they provide a useful framework for modelling temporal and spectral structure in speech patterns. Although it is clear that several properties of HMMs, such as their treatment of non-stationary patterns and their 'memoryless' nature, are inadequate in the full context of speech pattern modelling, these shortcomings are offset to some degree by the availability of mathematically sound and computationally efficient algorithms for automatic model parameter estimation from data and for pattern classification.

This paper reports on the results of a programme of comparative experiments using HMMs conducted at the Speech Research Unit, RSRE, between 1 September 1985 and 31 August 1986. The experiments examine the effect on the performance of a speaker dependent isolated digit recogniser of varying each of the principal parameters and algorithms of a particular class of HMM word-models.

HIDDEN MARKOV MODELS

The underlying assumption in the HMM approach to speech recognition is that a speech signal can be modelled as a *probabilistic function of a finite-state Markov process*. Given some representation $Y = Y(1),\ldots,Y(T)$ of an utterance as a sequence of vectors in d-dimensional space R^d, it is assumed that there is an underlying N state Markov chain $X = X(1),\ldots,X(T)$ such that Y is a random function of X. Intuitively the state sequence X corresponds to a sequence of high-level descriptors of the sounds in the utterance and the sequence Y is one of many possible acoustic realisations of X. The vector $Y(t)$ should be thought of as a description of some important characteristic of the acoustic signal at time t, for example a short-term power spectrum.

The Markov chain X is determined by the number of states N, an initial state probability vector $\pi = (\pi_1,\ldots,\pi_N)$, and a state-transition probability matrix $A = [a_{ij};\ i,j=1,\ldots,N]$, where,

$$\pi_j = prob(X(1) = s_j),\ j=1,\ldots,N,$$

$$a_{ij} = prob(X(t) = s_j \mid X(t-1) = s_i),\ i,j = 1,\ldots,N.$$

The pair $M = (\pi,A)$ is sufficient to characterise the statistical properties of the Markov chain X. M is called the *underlying Markov model*.

It remains to specify the relationship between the observed sequence Y and the hidden state sequence X. This is done by identifying each state s_i with a *probability density function* (pdf) b_i, such that

$$b_i(v) = prob(Y(t) = v \mid X(t) = s_i) \quad v \in R^d,\ i=1,\ldots,N.$$

The *hidden* Markov model $HMM = (\pi;A;b1,\ldots,bN)$ completely determines the stochastic process Y.

In the class of recogniser considered here a separate HMM is used to model each word in the vocabulary. Recognition is performed by comparing an unknown word with every HMM and assigning it to the class of the model which fits it most closely, in some sense. This raises two issues. First, how can an 'appropriate HMM' be constructed for a given word? Second, what measure should be used to determine how well a given HMM fits a particular unknown word-pattern?

All current solutions to the first problem involve two stages. The first stage, *initial parameter estimation*, consists of estimating the *structure* of an appropriate word-model. This estimate might be derived from a set of representative word-patterns, compiled using prior knowledge about the high-level structure of the word, or even chosen randomly. It is important not to underestimate the importance of this stage since essential properties of the final HMM, including the number of states *and* restrictions on the topology of the underlying Markov model, are fixed at this point.

The second stage is *parameter reestimation*. The parameters of the HMM are iteratively reestimated such that after each iteration the HMM is more representative of a set of training word-patterns, according to some criterion. Typically one of two closely related algorithms is used. The *forward-backward algorithm* increases the probability of the set of training patterns, conditioned on the HMM, at each iteration. An alternative is the *Viterbi reestimation algorithm*, which at each iteration increases the joint probability of the set of training patterns and the most probable underlying state sequence, given the HMM. At present these algorithms have only been validated for HMMs with particular classes of state output pdf b_i. These are the elliptically symmetric pdfs studied by Liporace [3], which include finite mixtures of multivariate gaussian pdfs, and "discrete" pdfs [4].

Once a HMM has been obtained for each word, an unknown word-pattern is classified in one of two ways. Either the probability of the word-pattern conditioned on each HMM is computed and the word-pattern is classified according to a *maximum likelihood* rule, or for each HMM the Viterbi algorithm is used to compute the joint probability of the word pattern and the most probable state sequence, and the pattern is classified according to this criterion.

In summary, in order to define an isolated word recognition system based on HMM word-models, the following issues must be addressed:

- the number of states in the underlying Markov source model,
- the topology of the underlying Markov source model,
- the class of the state output probability density functions,
- the amount of training data,
- the initial model parameter estimation algorithm,
- the model parameter reestimation algorithm, and
- the recognition algorithm.

The experiments reported below investigate the implications of particular choices of these parameters for recogniser performance.

SCOPE OF THE EXPERIMENTS

Speech Data

The speech data which was used for the experiments is a subset of the database described in [5]. In [5] 40 speakers are ranked according to the expected performance of a typical speaker-dependent isolated word recogniser on their speech. Each speaker in the study spoke 400 isolated digits (RSG10 random digit tables SB, 1A, 1B and 1C [6]), of which 100 were used for reference selection and the remaining 300 as test data. For the present experiments it was decided that only the speech of the 20 least consistent speakers from [5] would be used. For each speaker 100 digits were used as test data and subsets of the remaining 300 for training. The number of examples of each digit used for training was varied between 2 and 30 in the experiments. The data was digitised and preprocessed as described in [5].

HMM structure

The experiments were restricted to left-right HMMs and Gaussian states with diagonal covariance matrices. The number of states in the HMM was varied between 2 and 20.

Three types of topology were considered: *T1*, *T2* and *T3*. Figure 1 shows each of the topologies in the case of a 6-state model. *T1* is the minimal *left-right* topology which allows only transitions from state s_i to states s_i and s_{i+1}. *T2* includes all of the transitions permitted in *T1* plus transitions from state s_i to s_{i+2}. Topology *T3* is a full left-right topology in which transitions from state s_i to state s_j are allowed whenever $i < j$. These topologies are defined during initial parameter estimation and refined during reestimation. In particular, transitions which are initially permitted may vanish during the parameter reestimation process.

Gaussian pdfs with diagonal covariance matrix are the simplest members of the class of continuous states considered by Liporace [3]. Although they are not optimal [8], they avoid the need for computationally expensive matrix inversion, which is an important practical consideration. Furthermore, the application of the Viterbi recognition algorithm to this type of HMM is directly related to *dynamic time-warping* (DTW) using a weighted squared Euclidean distance measure [7], and so comparison with the results obtained using DTW gives a direct measure of the power of the formal HMM methods.

EXPERIMENTS IN SPEAKER-DEPENDENT ISOLATED DIGIT RECOGNITION

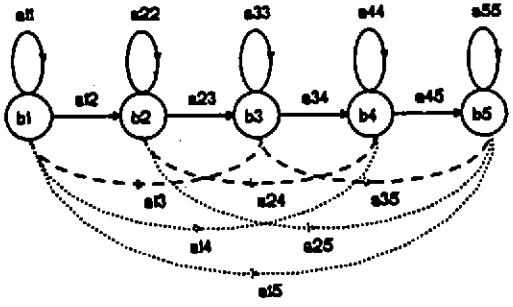


Figure 1: Topologies T1 (—), T2 (— &- - -) and T3 (— ,- - -&· · · · ·).

Initial HMM parameter estimation

In all of the experiments, initial HMM parameters were derived directly from the training data. Four alternative algorithms were considered:

IPE1: Composite reference, optimal segmentation,
IPE2: Composite reference, uniform segmentation,
IPE3: Optimal segmentation of individual word-patterns,
IPE4: Uniform segmentation of individual word-patterns.

IPE1 (Composite reference, optimal segmentation). This is a three stage algorithm. In the first stage all patterns representing a given word are combined to form a composite, or average, reference. The composite reference is then divided into N segments. Finally the segments are used to define initial estimates of the HMM state parameters.

Initially the first example of the word is chosen as the composite. Subsequent examples are aligned with the prototype composite, using a standard asymmetric DTW algorithm, and a new composite is obtained by averaging the prototype with the new word-pattern along the optimal time-registration path. The average is weighted to account for the number of words which have contributed to the composite. The DTW algorithm is defined by the recursive equation:

$$D(i,j) = min \{ D(i-1,j-x) + d(i,j) \} \quad (x=0,1,2),$$

where i and j index the frames of the new word-pattern and composite respectively and $d(i,j)$ is the euclidean distance between the ith frame of the word-pattern and the jth frame of the composite. The endpoints of the optimal time-registration path are constrained so that the first and last vectors in the composite are aligned with the first and last vectors in the individual word-pattern respectively.

The composite reference is then partitioned into N segments such that the fit, summed over all segments, between the vectors in the segment and a multivariate Gaussian pdf with the segment mean vector and diagonal covariance matrix is

maximised. This uses the optimal dynamic programming method described in [9].

The mean vector and covariance matrix of the ith segment are used as initial estimates for the corresponding parameters of the ith state of the HMM. The probability of a transition from state s_i to itself is set so that the expected duration of state s_i is the average number D_i of vectors per word-pattern contributing to the ith segment:

$$a_{ii} = (D_i-1)/D_i \quad (i=1,\ldots,N).$$

IPE2 (Composite reference, uniform segmentation). In *IPE2* a composite is formed as in *IPE1* and then partitioned uniformly into N equally long segments.

IPE3 (Optimal segmentation of individual word-patterns). In *IPE3* each pattern representing a given word in the training set is partitioned into N segments according to the optimal segmentation method described under *IPE1*. The mean vector and covariance matrix, computed over the ith segments in all of the segmented word-patterns, are used as initial estimates for the corresponding parameters of the ith state of the HMM. The probability of a transition from state i to itself is set so that the expected duration of state s_i is equal to the average length of the ith segment in the individual segmented patterns.

IPE4 (Uniform segmentation of individual word-patterns). *IPE4* is similar to *IPE3*, except that each pattern is partitioned uniformly into N equally long segments.

Parameter reestimation and recognition

The forward-backward algorithm and maximum likelihood classification, and Viterbi parameter reestimation and classification are both considered in the experiments. Unless otherwise stated the reestimation process was stopped either when the maximum permitted number of iterations (15) was reached, or the improvement after consecutive iterations fell below some fixed threshold (0.1).

RESULTS

It is not possible to present the results of all of the experiments in the space which is available and only those results which are considered to be of particular interest are discussed. The reader is refered to [10] for details. A subset of the results of the experiments is shown in figures 2 to 4. Figures 2 and 3 show results of experiments using initial parameter estimation algorithm *IPE1*, parameter reestimation by the forward-backward algorithm and maximum likelihood classification. The initial underlying Markov model topology is *T3* in figure 2 and *T1* in figure 3. Figure 4 shows the effect of different numbers of iterations of the forward-backward reestimation algorithm for the four initial parameter estimation algorithms in the study.

Number of states

Figure 2a shows recognition accuracy increasing with number of states independently of the size of the training set. This result is unexpected from the viewpoint of statistical parameter estimation. Intuitively as the number of states (and hence model parameters) is increased, more data is required for robust parameter estimation. Hence if the amount of training data is kept

constant and the size of the model is increased one would expect an eventual degradation in performance. The results suggest that this is offset by the ability of a HMM with a large number of states to model non-stationary regions more effectively.

The increase in error-rate as the number of states approaches 20 in figure 3a is due to the restrictive topology of *T1* and simply reflects the necessity of being able to 'skip' states in long models.

Size of training set

As one would expect, figures 2b and 3b show a general increase in performance as the amount of training data increases. This effect is most pronounced for HMMs with a large number of states, indicating that these are better able to take advantage of more generous amounts of training data.

Model topology

A comparison of figures 2a and 3a shows that the results for topologies *T3* and *T1* are comparable for models with up to 10 states. For longer models topology *T3* performs significantly better because of its ability to 'skip' states. This suggests that for short models a full left-right topology reduces to a more restricted form, like *T1* or *T2*, during reestimation, but that a more flexible topology is needed to take full advantage of a longer model.

This is confirmed by closer inspection of reestimated state transition probability matrices of HMMs with topology *T3*. In the case of speaker MW, for example, the percentage of states in the reestimated models which have non-zero probabilities assigned to transitions other than those permitted in *T1* is 0%, 12.5% and 62% for 4, 8 and 20 state models respectively. All transitions with non-zero reestimated probability in the 8 state models are permitted under topology *T2*, but this is not the case for the 20 state models in which non-zero probabilities were assigned to transitions which 'skip' two or more states.

Initial parameter estimation

Figure 4 shows percentage error as a function of number of iterations of the forward-backward algorithm for all four initial parameter estimation algorithms. The underlying Markov model has 8 states and topology *T2*.

Algorithm *IPE1* is least affected by reestimation: the error rate for *IPE1* is approximately 1% irrespective of the number of iterations of the forward-backward algorithm. This suggests that initial models defined by *IPE1* are close to local optima and hence not significantly changed by reestimation. The remaining algorithms all show an improvement after application of the forward-backward algorithm. For *IPE2* the error rate is reduced from 3% to 0.4% after 2 iterations and for *IPE3* and *IPE4* the reduction is from 1.25% and 0.7% respectively to 0.2% after 8 iterations. After reestimation the performance of all three algorithms is significantly better than that of *IPE1*.

The most surprising feature of the results is the superior performance, with or without parameter reestimation, of *IPE4* which is the least sophisticated of the four algorithms. The success of both *IPE3* and *IPE4* relative to the two algorithms which use a composite reference suggests that order-dependence in the construction of the composite is a serious problem. Alternative composite

reference algorithms are currently under consideration.

The wide variation in the performances of the algorithms reinforces the importance of reliable initial parameter estimation.

Reestimation and recognition algorithms

Experiments using the Viterbi reestimation and recognition algorithms are not yet completed. Results to date on models with up to 8 states have revealed no significant differences from the corresponding results for the forward-backward algorithm and maximum likelihood classification. Results on the Viterbi algorithm will be reported in full in [10].

CONCLUSIONS

This paper has presented a set of results which demonstrate the effect of particular algorithmic and parametric choices on the performance of a HMM based isolated word recogniser. Many of the results give quantitative evidence for expected effects; however there are also results, such as the relationship between model topology and model size and the performance of the initial parameter estimation algorithms, which are less predictable. The implications of these results will be investigated in future experiments.

REFERENCES

[1] G R Doddington and T B Smith, 'Speech recognition: turning theory to practice', IEEE Spectrum, 26-32, (September 1981).

[2] J K Baker, J M Baker, R Roth and P G Bamberg, 'Cost-effective speech processing', Proc. IEEE Int Conf ASSP, 9.7.1-9.7.4, (1984).

[3] L A Liporace, 'Maximum likelihood estimation for multivariate observations of Markov sources', IEEE Trans Information Theory, IT-28, 729-734, (1982).

[4] S E Levinson, L R Rabiner and M M Sondhi, 'An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition', Bell Syst Tech J, 62, 1035-1074, (1983).

[5] D C Smith, M J Russell and M J Tomlinson, 'Rank-ordering of subjects involved in the evaluation of automatic speech recognisers', RSRE memorandum 3926, (January 1986).

[6] R S Vonusa, J T Nelson, S E Smith and J G Parker, 'NATO AC/243 (Panel III RSG10) language data base', Proc NBS speech I/O standards workshop, (1982).

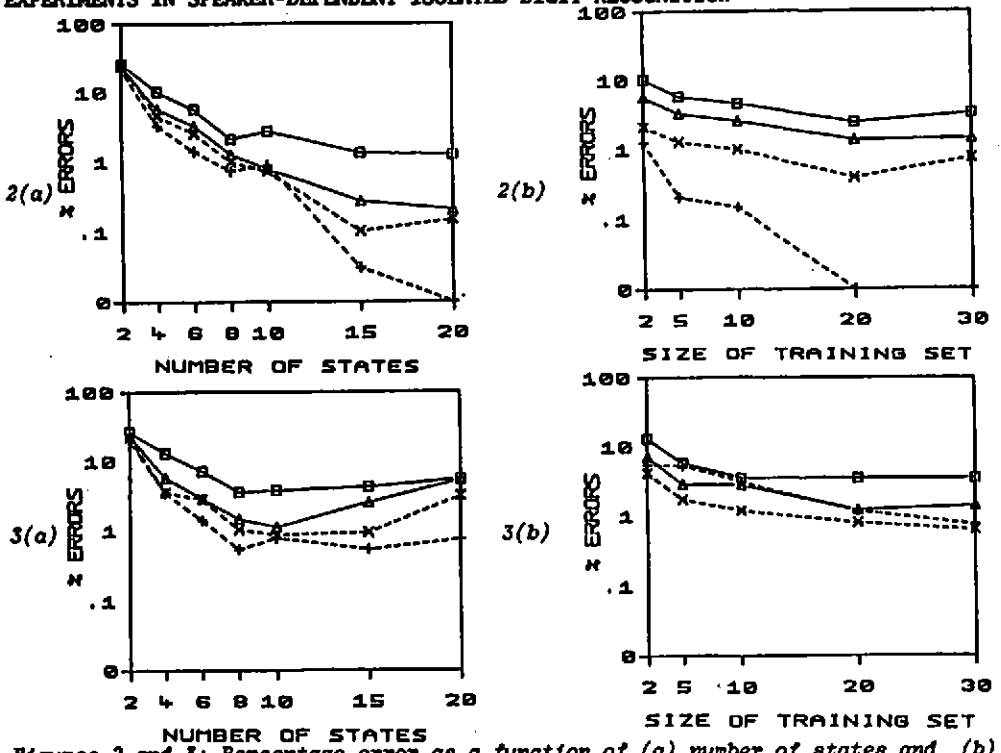
[7] J S Bridle, 'Stochastic models and template matching: some important relationships between apparently different techniques for automatic speech recognition', Proc Institute of Acoustics autumn conf. (1984)

[8] B H Juang, L R Rabiner, S E Levinson and M M Sondhi, 'Recent developments in the application of hidden Markov models to speaker independent isolated word recognition', Proc IEEE Int Conf ASSP, 1.3.1-1.3.4, (1985).

[9] J S Bridle and N C Sedgwick, 'A method for segmenting acoustic patterns, with applications to automatic speech recognition', Proc IEEE Int Conf ASSP, 656-659, (1977).

[10] M J Russell and A E Cook, 'Experiments in speaker-dependent isolated digit recognition using hidden Markov models: September 1985 - August 1986', RSRE memorandum, in preparation.

EXPERIMENTS IN SPEAKER-DEPENDENT ISOLATED DIGIT RECOGNITION



Figures 2 and 3: Percentage error as a function of (a) number of states and (b) size of training set (IPE1, forward-backward algorithm, maximum likelihood classification). Initial topology was T3 in figure 1 and T1 in figure 2. Figure (a) shows results for 2 (□), 5 (Δ), 10 (×) and 30 (+) examples of each digit for training. Figure (b) shows results for 4 (□), 6 (Δ), 8 (×) and 20 (+) state HMMs.

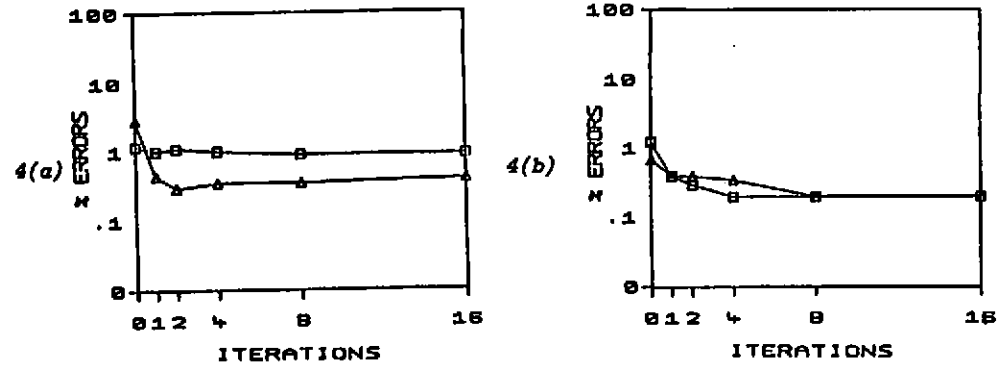


Figure 4: Percentage error as a function of number of iterations of the forward-backward algorithm for IPE1 ((a)□), IPE2 ((a)Δ), IPE3 ((b)□) and IPE4 ((b)Δ).