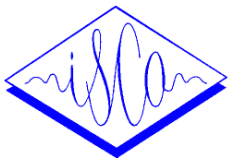

3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

AN EFFICIENT ALGORITHM TO FIND THE BEST STATE SEQUENCE IN HSMM

Antonio Bonafonte, Xavier Ros, Jose B. Mariño

Dept. of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC)
Apdo. 30002, Barcelona 08080, Spain
antonio@tsc.upc.es

ABSTRACT

Hidden Markov Modeling (HMM) techniques have been applied successfully to speech analysis. However, it has been claimed [1-7] that a major weakness of HMM is that the state duration probability density functions (SDPDF) are exponential, which is not appropriate for modelling speech events. In order to cope with this deficiency some authors have proposed to model explicitly the state duration. In these models the first order Markov hypothesis is broken in the loop transitions. Thus, the new models have been called Hidden Semi-Markov Models (HSMM). Different solutions have been proposed being the main common drawback the increase of the computational time by a factor D, being D the maximum time allowed in each state. In this paper a modified Viterbi algorithm which finds the best state sequence of HSMM is proposed. The proposed algorithm deals with log-convex parametric SDPDF. The log-convex property is fulfilled by the parametric functions usually applied. This method increases the computational burden with respect to conventional HMM by an empirical factor of just 3.2 without losing optimality and without increasing the storage with respect to other approaches. A more efficient algorithm is presented for the case that the duration of the states is modeled by bounded functions [7].

1.- INTRODUCTION

Many of the automatic speech recognition systems are based on Hidden Markov Models. These systems assume that duration of speech signal is well represented by exponential PDF. However, this assumption is not reasonable in speech modelling because it implies that the less time an event occurs, the most likely it is. As a consequence, HMM have been extended introducing an implicit or explicit model of the duration at each state.

The first idea, up to the authors knowledge, is due to Fergurson [1] and consists in explicitly define a probability function per state, p_i, which controls the occupancy in each state. In his paper, Fergurson estimated $p_i(d)$ from training data. One of the problems of this model is the large number of parameters per state (D, being D the maximum duration in any state). Those parameters have to be estimated in addition to those of the usual HMM. Therefore, an enormous database is required to accurately estimate the models. Fergurson himself suggested the possibility of using parametric functions for reducing the number of parameters.

Levinson [4] extended the Baum-Welch algorithm and proved its convergency for parametric HSMM. He also gave the details for the case of choosing the Gamma function as the PDF. Rusell and Moore [2] used the same result to recognize speech by means of a Poisson function. If the algorithm used for training the models is the segmental k-means then the same code can be used for training and recognition. In addition, the scaling problem can be ignored. In this context, Falachi [5] used a particular function chosen to increase the algorithm efficiency. Gu, Tseng and L. Lee [7] proposed the use of bounded functions (exponential functions lower and upper bounded) as a direct and simple (in training) but effective way of modelling the temporal structures existing in speech signals. They named their approach HMM/BSD (Hidden Markov Models with Bounded State Duration).

Ramesh and Wilpon [6] suggest the use of Inhomogeneus HMM (IHMM) as another way to reduce the number of parameters in the Fergurson model. It consists in defining variable state transition probabilities. Theoretically the number of parameter is even greater ($a_{ij}(d)$ has to be estimated for all i,j and d). However they propose to use a fixed value of $a_{jj}(d)$ for big values of d. Therefore, the number of parameters can be reduced.

Another way to treat the problem is to model implicitly the state duration by adding more states to a conventional HMM. The Fergurson model can be represented by a conventional HMM where each state is substituted by a subHMM of D states with tied observation probabilities. Rusell and Cook [3] proposed to substitute Fergurson submodels by smaller and more versatile versions (also with tied observation probabilities). The idea is to use a Markov chain in order to approximate any probability function. Furthermore, if the observation probabilities are not so strongly tied the result can derive to any of the specific chain topologies proposed in the literature, as for instance, the models proposed by K.F. Lee [8].

In this paper an efficient algorithm to find the best state sequence in HSMM is presented. In section 2 we review the computational burden of the approximations presented and state a theorem which can effectively reduce the complexity of these approximations. It is especially suitable to reduce complexity of HSMM as those proposed in [1,2,4,5]. In section 3, an appropriate algorithm for the use of bounded SDPDF is proposed. Section 4 is devoted to experimental results where the performance of different PDF is compared. Finally, some conclusions are reported.

This work was supported by the TIC grant number 92-10260-c02-02

2.- AN EFFICIENT ALGORITHM

The main drawback of HSMM is the increase of computational time that they report. A naive implementation of a modified Viterbi algorithm can increase it about $D^2/2$ times and besides that, the storage is multiplied by D [1,6]. This enormous burden has been reduced by using a particular kind of density functions [5] or Inhomogeneous HMM [6]. The algorithm proposed in [6] for IHMM has the same complexity than the proposed in [7] for any (parametric or not) distribution function. This algorithm is D times more expensive than conventional Viterbi. However, the proposal of [7] also increases by two the memory required by the first proposal [1].

The version we have developed has the same computational requirement than the proposed in [7] without any increase of the storage capacity. The main recursion of the modified Viterbi algorithm follows:

$$\delta_t(i) = \max_{\tau} \left\{ \beta(\tau)\, p_i(t-\tau) \max_{j} \{\delta_\tau(j)\, a_{ji}\} \right\} \quad (1)$$

where

$\delta_t(j)$ is the probability of the best state sequence observing O_t at state j (and O_{t+1} in another state),

$p_j(d)$ is the probability of observing d symbols at state j and

$\beta(\tau) = \prod_{k=\tau+1}^{t} b_j(O_k)$ which can be computed recursively from $\beta(\tau+1)$ when maximizing (1)

Although this version is far away from the first implementations, mainly due to the definition of $\beta(\tau)$, a factor of D with respect to conventional HMM is still an important drawback. In this section we study a property that is desirable in the SDPDF so that the computational effort can be effectively reduced.

Let hypothesis H1 be the best path leaving state j_1 at time τ_1. Analogously, hypothesis H2 can be defined. Suppose this two hypotheses are competing in order to calculate $\delta_t(i)$ and H1 is the winner. It would be interesting to know under which conditions H1 is going to conserve the leadership when calculating $\delta_{t'}(i)$, with $t' > t$. Then a pruning theorem could be stated in order to avoid the progress of H2 when computing $\delta_{t'}(i)$ if these conditions are accomplished. Let us give it in a proper form.

• H1 is the winner when computing $\delta_t(i)$

$$\frac{\delta_{\tau_1}(j_1) \prod_{k=\tau_1+1}^{t} b_i(O_k)\, p_i(t-\tau_1)}{\delta_{\tau_2}(j_2) \prod_{k=\tau_2+1}^{t} b_i(O_k)\, p_i(t-\tau_2)} = K > 1 \quad (2)$$

• Will H1 keep its top position when calculating $\delta_{t'}(i)$?

$$\frac{\delta_{\tau_1}(j_1) \prod_{k=\tau_1+1}^{t'} b_i(O_k)\, p_i(t'-\tau_1)}{\delta_{\tau_2}(j_2) \prod_{k=\tau_2+1}^{t'} b_i(O_k)\, p_i(t'-\tau_2)} = K \cdot \frac{p_i(t'-\tau_1)\,/\,p_i(t-\tau_1)}{p_i(t'-\tau_2)\,/\,p_i(t-\tau_2)} \overset{?}{>} 1 \quad (3)$$

In the most critical case suppose $K \approx 1$. Computing the logarithms in (3) and defining $g_i(x)=\log(p_i(x))$ the inequation is transformed as follows:

$$g_i(t'-\tau_1) - g_i(t-\tau_1) \overset{?}{\geq} g_i(t'-\tau_2) - g_i(t-\tau_2) \quad (4)$$

Denoting $d_1 = t-\tau_1$, $d_2 = t-\tau_2$ and being $d = t' - t > 0$,

$$g_i(d+d_1) - g_i(d_1) \overset{?}{\geq} g_i(d+d_2) - g_i(d_2) \quad (5)$$

If $g_i'(d_1) > g_i'(d_2)$ and $g_i'(x)$ is monotonous in an interval containing d_1, $d+d_1$, d_2 and $d+d_2$ then the inequality (5) is true ((5) is also true if $g_i'(x)$ is constant in this interval). In particular, eq. (5) holds in the following cases:

- $d_1 > d_2$ $(\tau_1 < \tau_2)$ and $g_i'(x)$ monotocally increasing, so $g_i''(x) \geq 0\ \forall x$
- $d_1 < d_2$ $(\tau_1 > \tau_2)$ and $g_i'(x)$ monotocally decreasing, so $g_i''(x) \leq 0\ \forall x$

Obviously, if $d_1 = d_2$ $(\tau_1 = \tau_2)$ the inequalities become equalities and H1 will always be more favorable than H2.

Definition:

A function p(x) is said to be log-convex if $[\log(p(x))]'' \leq 0\ \forall x$
A function p(x) is said to be log-concave if $[\log(p(x))]'' \geq 0\ \forall x$

Pruning theorem:

i) if the SDPDF of the state i is log-convex then (see figure I) if the best path (path 1) leaving state i at time t has arrived at state i at time τ from state j then the best path (path 2) leaving state i at time t+1 has arrived at state i at:

- time τ from state j
- time τ' from any state, with $\tau' > \tau$

ii) if the SDPDF of the state i is log-concave then if the best path (path 1) leaving state i at time t has arrived at state i at time τ from state j then the best path (path 2) leaving state i at time t+1 has arrived at state i at:

- time τ from state j
- time τ' from any state, with $\tau' < \tau$
- time t from any state: these paths did not compete when calculating $\delta_t(i)$

Consequence: if the SDPDF of the state i is log-concave and log-convex, $([\log(p_i(x))]'' \equiv 0\ \forall x$, that is, $p_i(x)$ is a exponential PDF) then the best path (path 2) leaving state i at time t+1 has arrived at state i at:

- time τ from state j
- time t from any state: these paths did not compete when calculating $\delta_t(i)$

The classical Viterbi algorithm relies on this consequence.

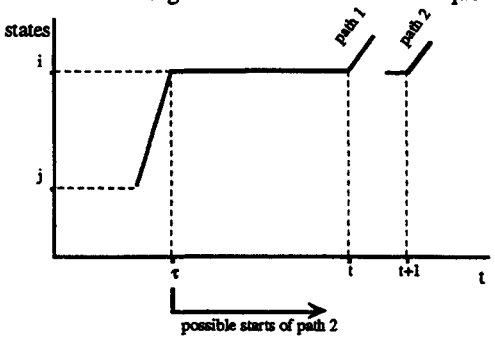

Figure I

The application of this property to the modified Viterbi algorithm is quite direct. Classically, the parameter τ of the recursion (1) goes from t-D to t-1. If function $p_i(d)$ is log-convex then the lower limit can be set to τ^* according to the pruning theorem without losing of optimality. Most parametric functions used by HSMM are log-convex. For instance the one-side normal, Rayleigh, Maxwell and Poisson functions are log-convex. The duration function proposed by Falachi is also log-convex. Gamma function is log-convex if the mean is greater than the standard deviation. This is the situation we have found in all cases during the recognition stage. However, this relation can be violated at some state during training. In this case, the function would be log-concave and not the lower, but the upper limit of the recursion can be optimized.

In order to compare the efficiency of this approach we have studied $E[t-\tau^*]$ and $\max[t-\tau^*]$. As D should be greater than $\max[t-\tau^*]$ the efficiency gain with respect to eq. (1) is greater than $G = \max[t-\tau^*]/E[t-\tau^*]$. The value we have found for G is 6.2 in the experiments we have performed with the gamma function. In our application, values of 20 for D are large enough. This leads to an increase of computational effort of around 3.2 times with respect to conventional HMM, (far away from $D^2/2 = 200$).

Furthermore, the computational time is almost independent of D. As a consequence, the choice of this parameter is not so critical because it only influences the memory requirements but not the complexity.

An even greater efficiency could be obtained if information of the second candidate was stored. In this case, the value of τ' in the expressions $\tau' < \tau$ and $\tau' > \tau$ in the pruning theorem could be changed to the time when the second candidate left the previous state. However, the incorporation of this information is not so direct and the storage requirements increase.

3.- HMM WITH BOUNDED STATE DURATION

HMM/BSD are detailed in [7]. Basically, in HMM/BSD the duration of each state i is upper and lower bounded by two parameters, u_i and l_i. The recursion (1) is also valid. However, the τ recursion goes from t-u_i to t-l_i. If bounded functions are chosen to be exponential (as proposed in [7]) then the SDPDF are the same than those of conventional HMM except that the lower and the upper parts of the exponentials have been removed. In this case, if two candidates with allowed durations were compared when computing $\delta_t(i)$, the result of the comparison will be exactly the same when computing $\delta_{t+1}(i)$ if the durations of the two paths are still allowed. The search space for computing $\delta_t(i)$ and $\delta_{t+1}(i)$ can be divided in three regions (figure II). Region A contains the path origins which are possible in t but not in t+1 (because the length would be longer than u_i in t+1). Its width is 1. Analogously, region C contains the path origins which are possible in t+1 but not in t. Its width is also 1. Finally, region B contains the possible path origins which are possible in both, t and t+1. Its width is $u_i - l_i$. The algorithm is driven as follows:

Case 1: if the duration of the optimum path when computing $\delta_t(i)$ is u_i, (it belongs to region A) then it cannot be considered when calculating $\delta_{t+1}(i)$ and all the search space (regions B and C) must be explored.

Case 2: if the duration of the optimum path when computing $\delta_t(i)$ is smaller than u_i, (it belongs to region B) then only paths which were not used when computing $\delta_t(i)$ (paths belonging to region C) are possible rivals. In this case the number of possible candidates is the same that with conventional Viterbi.

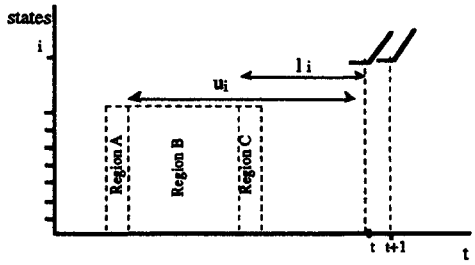

Figure II

Let f_{1i} (f_{2i}) be the occurrence ratio of case 1 (case 2) at state i. The increase of computation with respect to conventional Viterbi is around

$$E[f_{1i} \cdot 3\ (u_i - l_i + 1) + f_{2i} + l_i/N']$$

where N' stands for the mean number of predecessor states. The third term of the addition is due to the cost of the last l_i observations which are common to all the paths arriving at state i, that is,

$$(a_{ii})^{l_i} \cdot \prod_{k=t-l_i+1}^{t} b_i(O_k)$$

As f_{1i} is much lower than f_{2i}, the computational effort is similar to conventional Viterbi.

4. EXPERIMENTAL RESULTS

In this section some preliminary speech recognition experiments are presented. The objective of these experiments is to estimate the gain efficiency given in section 2 and to verify the behavior of HSMM with different state duration probability density families.

The experiment consists in the recognition of the catalan digits. Ten speakers uttered each digit ten times. Five speakers were used for training and the other five were reserved for test. The preprocessing stage is detailed in [9]. Basically it consist of 12 LPC-cepstral, 12 δ-cepstral and one δ-energy coefficients. Five states discrete left to right HMM have been used.

The algorithm proposeded in section 2 has been used during the training and the recognition stages. The segmental k-means algorithm is modified as follows: after the modified Viterbi algorithm finds the best state sequence, not only the A and B parameters, but also the mean and the standard deviation of each state duration PDF are updated. For the SDPDF that we have chosen these two parameters determine the PDF.

In table I the number of errors obtained for four different functions is presented. These four functions have also been tested normalizing the state duration by the overall word duration in order to normalize different speech rates. This duration normalization, proposed in [7] does not improve very much the error rate, surely because it ignores the different durations of the words. Nevertheless, this normalization would have been difficult to extend to the case of continuous speech.

	classical HMM	Poisson	Maxwell	Rayleigh	Gamma
not norm.	10	5	2	5	1
norm.	–	4	4	7	1

Table I. Number of errors versus SDPDF for normalized and not normalized durations. 500 digits were recognized.

The results show that the best performance is obtained with the Gamma function. Furthermore, the improvement obtained using HSMM is illustrated.

Bounded functions:

The bounding parameters of the HMM/BSD, as proposed in [7], are estimated in the training phase from the best state sequence obtained by the conventional Viterbi. However, we have found that better results are obtained if the bounded SDPDF are also used for training. In the experiments we have performed, the minimum (m_i) and maximum (M_i) duration at each state is computed at each iteration of the training algorithm. Afterwards l_i and u_i are set to 0.75 m_i and to 1.5 M_i respectively. Table II shows the results obtained if the SDPDF are bounded exponential or bounded uniform functions. As in the results presented before, normalization by the overall duration of the whole word has been tried. It can be seen that in this case it improves the recognition rate. A reason can be the abrupt boundaries which can produce an error if the test utterance is slower or faster than the training utterances.

	exponential only recog.	uniform	exponential
not normal.	10	4	4
normalized	9	2	3

Table II. Number of errors using bounded functions. a) only in recognition, b) and c) in training and recognition. 500 digits were recognized.

As suggested for classical HMM [10], the use of exponential functions does not improve the performance with respect to uniform functions.

CONCLUSIONS

In this paper an efficient algorithm has been proposed to find the best state squence through a HSMM network. The algorithm is specially convenient when the second derivative of the logarithm of the PDF is always positive or always negative. This condition is accomplished by most of the PDF currently used in speech modelling. Another algorithm has been introduced in the case of having the duration modelled by a bounded exponential function.

Some isolated speech recognition experiments have been performed to compare different PDF. Gamma function gives the best results. Bounded functions with normalized duration also provide very good results. As the computational effort that HMM/BSD require is smaller than in the other approaches, they represent a good option to model the duration of the speech events.

We have done preliminary experiments with continuous speech but the results using classical k-means are much worse than the ones obtained training with the Baum-Welch algorithm. Therefore, a large database should be used or a subsequent reestimation of the observation probabilities by the Levinson algorithm [4] should be done.

REFERENCES

[1] J.D.Fergurson, "Variable Duration Models for Speech", Proc. Symposium. on the Application of Hidden Markov Models to Text and Speech, pp. 143-179, Oct. 1980

[2] M.J.Russell and R.K.Moore, "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition," in Proc. ICASSP'85 (Tampa,FL), pp.5-8, Mar. 1985

[3] M.J.Russell and A.E.Cook, "Experimental Evaluation of duration modelling techniques for Automatic Speech Recognition," in Proc. ICASSP'87, pp.2376-2379.

[4] S.E.Levinson, "Continuously variable duration hidden Markov models for automatic speech recognition," Computer, Speech and Language, vol 1, pp. 29-45, Mar 1986

[5] A.Falaschi, "Continuously Variable Transition Probability HMM for Speech Recognition," in Speech Recognition and Understanding, P.Laface and R. De Mori, Ed. Springer-Verlag Berlin Heidelberg, 1992, pp. 125-130.

[6] P.Ramesh and J.G.Wilpon, "Modeling state durations in hidden Markov models for automatic speech recognition," in Proc. of ICASSP'92 (San Francisco, CA), pp. 381-384.

[7] H. Gu, C. Tseng and L. Lee, "Isolated-Utterance Speech Recognition Using Hidden Markov Models with bounded State Duration," IEEE Trans. on Signal Processing, vol. 39, no. 8, pp. 1743-1751, Aug. 1991

[8] K.F. Lee, "Automatic Speech Recognition. The development of the SPHINX System", Kluwer Academic Publishers,1989

[9] J.B.Mariño, A.Bonafonte, A.Moreno, E.Lleida, C.Nadeu, E.Monte, "Recognition of Numbers by Using Demisyllables and Hidden Markov Models," Proc. of Fifth European Signal Processing Conference 90, (Barcelona), pp. 1363-1366.

[10] L. Rabiner, B.H. Juang, "Hidden Markov Models for Speech Recognition - Strenghts and Limitations," in Speech Recognition and Understanding, P.Laface and R. De Mori, Ed. Springer-Verlag Berlin Heidelberg, 1992, pp. 3-29.