Automatic Word Recognition Based on Second-Order Hidden Markov Models

Jean-Francois Mari, Jean-Paul Haton, *Fellow, IEEE*, and Abdelaziz Kriouile

***Abstract*— We propose an extension of the Viterbi algorithm that makes second-order hidden Markov models computationally efficient. A comparative study between first-order (HMM1's) and second-order Markov models (HMM2's) is carried out. Experimental results show that HMM2's provide a better state occupancy modeling and, alone, have performances comparable with HMM1's plus postprocessing.**

I. Introduction

HMM-BASED speech modeling assumes that the input signal can be split into segments modeled as states of an underlying Markov chain and that the waveform of each segment is a stationary random process. In a first-order hidden Markov Model (HMM1), the sequence of states is assumed to be a first-order Markov chain. This assumption is mainly motivated by the existence of efficient and tractable algorithms for model estimation and recognition proportional to N^2T, where N is the number of states of the model and T the utterance length. However, HMM1's suffer from several drawbacks. For instance, in an HMM1, the frames inside a segment are assumed to be independent, and trajectory modeling (i.e., frame correlation) in the frame space is not included. By incorporating short-term dynamic features to model spectrum shape, an HMM1 can be made to overcome this drawback. Modeling segment duration, which follows a geometric law as a function of time, remains another major drawback of the HMM1. In this paper, we investigate models where the underlying state sequence is a second-order Markov chain (HMM2), and we study their capabilities in terms of duration and frame correlation modeling. Because the major disadvantage of our technique is the computational complexity, we propose an appropriate implementation of the re-estimation formulas that yields algorithms only N_i times slower compared to the HMM1 ones, where N_i is the average input branching factor of the model. We show that HMM2's overcome HMM1's, but the offset between performances is significantly reduced after a post-processing in which durational constraints based on states occupancy are incorporated to an HMM1-based recognizer. This paper is organized as follows. Section II introduces second-order Markov chains, and Section III summarizes the Viterbi and Baum–Welch algorithms for the HMM2. In Section IV, we discuss the complexity of the HMM2. Section V is devoted to the postprocessor that implements state duration constraints. The experimental comparison is described in Section VI, and we draw some conclusions in Section VII.

Manuscript received July 20, 1992; revised July 30, 1996. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xuedong Huang.

The authors are with CRIN/CNRS and INRIA-Lorraine, Bâtiment LORIA, BP 239, Vandoeuvre les Nancy, 54506 cedex, France (e-mail: jfmari@loria.fr).

Publisher Item Identifier S 1063-6676(97)00764-5.

II. Second-Order HMM's

Unlike the first-order Markov chain where the stochastic process is specified by a 2-D matrix of *a priori* transition probabilities $\{a_{ij}\}$ between states s_i and s_j, the second-order Markov chain is specified by a 3-D matrix $\{a_{ijk}\}$.

$$\begin{aligned} a_{ijk} &\triangleq \text{Prob}(q_t = s_k / q_{t-1} = s_j, q_{t-2} = s_i, q_{t-3} = \cdots) \\ &= \text{Prob}(q_t = s_k / q_{t-1} = s_j, q_{t-2} = s_i) \end{aligned} \quad (1)$$

with the constraints

$$\sum_{k=1}^{N} a_{ijk} = 1 \text{ with } 1 \le i \le N, 1 \le j \le N$$

where N is the number of states in the model, and q_t is the actual state at time t.

The probability of the state sequence $Q \triangleq q_1, q_2, \cdots, q_T$ is defined as

$$\text{Prob}(Q) = \Pi_{q_1} a_{q_1 q_2} \prod_{t=3}^{T} a_{q_{t-2} q_{t-1} q_t} \quad (2)$$

where Π_i is the probability of state s_i at time $t = 1$, and a_{ij} is the probability of the transition $s_i \to s_j$ at time $t = 2$.

Each state s_i is associated with a mixture of Gaussian distributions

$$b_i(O_t) \triangleq \sum_{m=1}^{M} c_{im} \mathcal{N}(O_t; \mu_{im}, \Sigma_{im}), \text{with} \sum_{m=1}^{M} c_{im} = 1 \quad (3)$$

where O_t is the input vector (the frame) at time t.

Given a sequence of observed vectors $O \triangleq O_1, O_2, \cdots, O_T$, the joint state-output probability $\text{Prob}(Q, O/\lambda)$ is defined as

$$\begin{aligned} &\text{Prob}(Q, O/\lambda) \\ &\quad = \Pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \prod_{t=3}^{T} a_{q_{t-2} q_{t-1} q_t} b_{q_t}(O_t). \end{aligned} \quad (4)$$

Each second-order Markov model has an equivalent first-order model on the twofold product space $\mathbf{S} \times \mathbf{S}$, but going back to first order increases dramatically the number of states. For instance, Fig. 2 shows the equivalent HMM1 associated with

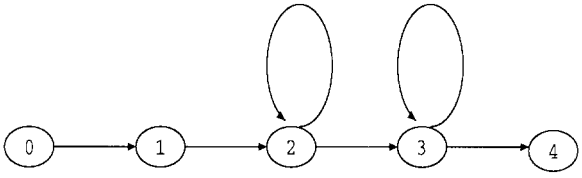

Fig. 1. Original second-order model.

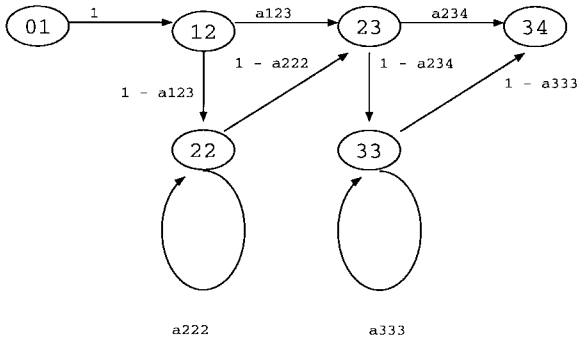

Fig. 2. First-order equivalent model.

the HMM2 depicted in Fig. 1. In this model, the states in the same column share the same pdf.

This topology has already been investigated in several works [5]–[7]. It is interesting to note that HMM2 converge naturally to Ferguson-like models.

In the model depicted in Fig. 2, the duration in state j may be defined as

$$\begin{aligned} &d_j(0) = 0 \\ &d_j(1) = a_{ijk}, i \neq j \neq k \\ &d_j(n) = (1 - a_{ijk}) \cdot a_{jjj}^{n-2} \cdot (1 - a_{jjj}), n \geq 2. \end{aligned}$$

The state duration in a HMM2 is governed by two parameters, i.e., the probability of entering a state only once, and the probability of visiting a state at least twice, with the latter modeled as a geometric decay. This distribution better fits a probability density of durations [2] than the classical exponential distribution of an HMM1. This property is of great interest when a HMM2 models a phoneme in which a state captures only one or two frames.

III. Extended Viterbi and Baum–Welch Algorithms

The extension of the Viterbi algorithm to HMM2 is straightforward. We simply replace the reference to a state in the state space $\mathbf{S}$ by a reference to an element of the twofold product space $\mathbf{S} \times \mathbf{S}$. The most likely state sequence is found by using the probability of the partial alignment ending at transition (s_j, s_k) at times $(t-1, t)$

$$\begin{aligned} &\delta_t(j,k) \triangleq \text{Prob}(q_1, \cdots q_{t-2}, q_{t-1} = s_j, q_t = s_k, O_1, \cdots, O_t/\lambda) \\ &\quad 2 \leq t \leq T, \;\; 1 \leq j,k \leq N. \end{aligned} \tag{5}$$

Recursive computation is given by

$$\begin{aligned} &\delta_t(j,k) = \max_{1 \leq i \leq N}[\delta_{t-1}(i,j) \cdot a_{ijk}] \cdot b_k(O_t) \\ &\quad 3 \leq t \leq T, \;\; 1 \leq j,k \leq N. \end{aligned} \tag{6}$$

To adjust the second-order HMM parameters, we define the new forward and backward functions on which the extended Baum–Welch algorithm is based. The forward function $\alpha_t(j,k)$ defines the probability of the partial observation sequence $O_1, \cdots, O_t$ and the transition (s_j, s_k) between time $t-1$ and t

$$\begin{aligned} &\alpha_t(j,k) \triangleq \text{Prob}(O_1, O_2, \cdots, O_t, q_{t-1} = s_j, q_t = s_k/\lambda) \\ &\quad 2 \leq t \leq T, \;\; 1 \leq j,k \leq N. \end{aligned} \tag{7}$$

As in HMM1, $\alpha_t(j,k)$ can be computed from $\alpha_{t-1}(i,j)$ in which (s_i, s_j) and (s_j, s_k) are two transitions between states s_i and s_k

$$\begin{aligned} &\alpha_{t+1}(j,k) = \sum_{i=1}^{N} \alpha_t(i,j).a_{ijk}.b_k(O_{t+1}) \\ &\quad 2 \leq t \leq T-1, \;\; 1 \leq j,k \leq N. \end{aligned} \tag{8}$$

Similarly, the backward function $\beta_t(i,j)$, which is defined as the probability of the partial observation sequence from $t+1$ to T, given the model λ and the transition (s_i, s_j) between times $t-1$ and t, can be expressed as

$$\begin{aligned} &\beta_t(i,j) \triangleq \text{Prob}(O_{t+1}, \cdots O_T/q_{t-1} = s_i, q_t = s_j, \lambda) \\ &\quad 2 \leq t \leq T-1, \;\; 1 \leq i,j \leq N. \end{aligned} \tag{9}$$

IV. Implementation and Complexity

A naive implementation of the recursion for the computation of α and β requires on the order of N^3T operations, compared with N^2T for the standard HMM1. A more precise analysis to (8) shows that the sum is taken over all nonzero transitions a_{ijk} for a given pair of successive states (j,k). Therefore, the algorithm implements a table of list heads that associates the list of transitions a_{ijk} ending at each state k

$$k \rightarrow (i_0, j_0, k, a_{i_0 j_0 k}), (i_1, j_1, k, a_{i_1 j_1 k}), \cdots$$

If N_i is the average input branching factor of the model, this list requires N_i more memory space compared with the equivalent list in HMM1. Similarly, the computation of β requires the implementation of lists of transitions accessed by a pair of states

$$(i,j) \rightarrow (i, j, k_0, a_{ijk_0}), (i, j, k_1, a_{ijk_1}), \cdots$$

This table of list heads whose size is N. N_i replaces the table that usually implements the access given a state to its list of successors for the purpose of α and β computation. Therefore, in HMM2, we have N_i more lists with an average of N_i successors. Of course, α is represented by a N^2T matrix, whereas β needs only a N^2 matrix. These remarks suggest that, in theory, there is roughly a factor of N_i in terms of space and computation requirements between HMM1 and HMM2. In our system, the average input branching factor is 2, and the re-estimation and recognition algorithm are less than two times slower, which is still tractable. In fact, experimental measures on CPU times of both systems have shown that HMM2's are only 1.5 slower than HMM1's since the system spends half of its time in computing the likelihood of frames given a state output pdf with full covariance matrices than accessing the lists of successors.

V. Duration Model

Even if the duration of a segment is better modeled by two parameters in an HMM2, thus avoiding singular state assignment as previously mentioned, it is nevertheless necessary to implement duration constraints based on the relative duration of the segments corresponding to successive states, as in [7]. The reason is that most of the errors of our HMM2-based word recognition system come from singular alignments given by the Viterbi algorithm. We observed that state durations were strongly correlated for states in a model. In order to take this correlation into account, we have specified a set of classes of correct alignments on a one class per model basis. Given an utterance, an alignment between a model and the utterance is defined by a vector of relative duration of the states of the model. This alignment is found using the Viterbi algorithm. We denote the following:

$\mathbf{w}$ d-frame-long word that has been aligned with HMM λ. Each state i among the N states of λ captures d_i frames. If all states must be visited, we have $d = d_1 + d_2 + \cdots + d_N$

$$x \triangleq (d, \frac{d}{d_2}, \frac{d}{d_3}, \cdots, \frac{d}{d_N}) \tag{10}$$

$\mathbf{g}_\lambda$ mean vector associated with the class of λ and V_λ, which is the covariance matrix,

$\boldsymbol{detV_\lambda}^{1/N}$ normalizing factor that ensures that all matrices have a determinant equal to one.

Given a word model and the class of correct alignments of this model, we measure the distance between alignments using the Mahalanobis distance

$$d^2(x, g_\lambda) \triangleq \det V_\lambda{}^{1/N} (\mathbf{x} - \mathbf{g}_\lambda)^t \mathbf{V}_\lambda{}^{-1} (\mathbf{x} - \mathbf{g}_\lambda). \tag{11}$$

This distance weights the probability of the Viterbi's alignment during a postprocessing step, where the N-best[1] answers given by the recognition algorithm [6] are rescored.

$$\text{Final Score} = A \cdot d^2(x, g_\lambda) + B \cdot \log(P(O/\lambda)). \tag{12}$$

A and B are normalizing constants determined empirically on the training set.

VI. Experimental Comparison

A. Test Protocol

First-order HMM's and second-order HMM's have been comparatively assessed using the same database of digits, i.e., the adult part of the TI-NIST database. The vocabulary is made up of 23 models—one per digit and gender—and one for the background noise. The state output densities are mixtures of nine Gaussian estimates with full covariance matrices. For comparison, we have used models with the same topology and the same number of pdf's. In particular, digit models have six states with five self loops and no skip transition, whereas the background noise model has only two states and one self loop.

[1] N does not refer to the number of states of a model but rather to the number of alignments.

TABLE I
String Error Rates (without Post-processing)

Parameterization	Male + Female	
	HMM1	HMM2
$11MFCC + 11\Delta + \Delta E + \Delta\Delta E$	4.5% (4.1 5.0)	2.4% (2.1 2.7)
$11MFCC + 11\Delta + \Delta E + \Delta\Delta E + 11\Delta\Delta$	3.7% (3.3 4.1)	2.4% (2.1 2.7)

TABLE II
String Error Rates (with Post-processing)

Parameterization	Male + Female	
	HMM1	HMM2
$11MFCC + 11\Delta + \Delta E + \Delta\Delta E$	2.8% (2.5 3.2)	2.2% (1.9 2.5)
$11MFCC + 11\Delta + \Delta E + \Delta\Delta E + 11\Delta\Delta$	2.3% (2.0 2.6)	2.1% (1.8 2.4)

TABLE III
Comparison Between HMM1 (with Post-processing) and HMM2 (without Post-processing)

	HMM1	HMM2
Insertions	174	159
Deletions	14	20
Substitutions	31	34
String error rate	2.3 %	2.4 %
% correct	99.8	99.8
Accuracy	99.2	99.2

B. Acoustic Analysis

Using a frame shift of 12 ms and a 25-ms window, 12 cepstral coefficients corresponding to an approximate Mel-frequency warped spectrum are computed. The first coefficient, which is called loudness, was removed. Two analysis feature vectors incorporating dynamic features have been specified in order to explore the capability of HMM2's to capture frame correlations.

- 24 coefficients: 11 static, 12 dynamic first-order coefficients plus the second-order energy coefficient $\Delta\Delta E$.
- 35 coefficients: 11 static, 12 dynamic first-order coefficients plus 12 dynamic second-order coefficients.

C. Comparison HMM1/HMM2

Tables I and II summarize the recognition results. In these tables, the string error rates and the corresponding 95% confidence intervals are given. Table III gives the results at the word level. In the different experiments, we used the 8700 strings from the test part of the TI-NIST database containing 28 383 digits. Throughout these experiments, a grammar of unknown length strings was used.

Three major conclusions can be drawn from these results:

1) HMM2 outperforms HMM1 in the absence of postprocessing, and HMM2 without postprocessing is almost equivalent in performance to HMM1 with postprocessing.
2) Acceleration coefficients do not significantly improve performance, especially with HMM2.

3) The offset in performances is greatly reduced when a postprocessor is used to take into account the duration constraints.

Point 1 can be explained by the capability of HMM2 to explicitly model the event that a state can be visited just one time and eliminate singular alignments given by the Viterbi algorithm in the recognition process when a state captures just one frame whereas all other speech frames fall into the neighboring states. Thus, the trajectory of speech, in terms of state sequence, is better modeled by HMM2. Point 2 has already been mentioned in relation to clean speech and HMM1 models [4]. Since the beginning of this study in 1990, several sytems have produced better performances on the TI-NIST corpus [3], [1]. These systems involve sophisticated acoustic analysis and training techniques. Our word recognition system, based on HMM1 models, which serves as the reference system to which the HMM2-based system was compared, gives results similar to the system described by Wilpon in 1993 [8], i.e., 2.4% string error rate with a ten-state model with nine Gaussian pdf per mixture and telephone bandwith speech. In our system, we have six states per model but two models per digit. This keeps the number of parameters slightly constant.

VII. Conclusion

We have described a connected word recognition system based on second-order Markov models. We have presented the capability of HMM2 to capture some duration constraints at the state level. Usually, these duration constraints are implemented in postprocessors that are trained separately. The maximum likelihood principle in the training is thus not guaranteed, whereas HMM2's converge with the forward-backward algorithm naturally to Ferguson-like models in which the state duration is represented by a nonparametric pdf for small values and a geometric law for higher values. Besides, the duration modeling with parametric pdf's in semi-Markov process results in a significant increase in computational complexity, whereas a second-order Markov process gives a crude but tractable answer.

References

[1] R. Cardin, Y. Normandin, and E. Millien, "Inter-word coarticulation modeling and mmie training for improved connected digit recognition," in *Proc. IEEE-ICASSP*, vol. 2, 1993, pp. 243–246.
[2] T. H. Crystal and A. S. House, "Segmental durations in connected speech signals: Current results," *J. Acoust. Soc. Amer.*, vol. 83, no. 4, pp. 1553–1573, Apr. 1988.
[3] R. Haeb-Umbach, D. Geller, and H. Ney, "Improvements in connected digit recognition using linear discriminant analysis and mixture densities," in *Proc. IEEE-ICASSP*, 1993, pp. 239–242.
[4] B. A. Hanson and T. Applebaum, "Robust speaker-independent word recognition using static, dynamic, and acceleration features: Experiments with Lombard and noisy speech," in *Proc. IEEE-ICASSP*, 1990, pp. 857–860.
[5] M. J. Russell and A. Cook, "Experimental evaluation of duration modeling techniques for automatic speech recognition," in *Proc. IEEE-ICASSP*, 1987, pp. 2376–2379.
[6] R. Schwartz and S. Austin, "A comparison of several approximate algorithms for finding multiple (N-BEST) sentence hypotheses," in *Proc. IEEE-ICASSP*, 1991, pp. 701–704.
[7] N. Suaudeau and R. André-Obrecht, "Sound duration modeling and time variable speaking rate in a speech recognition system," in *Proc. Eurospeech*, 1993, pp. 307–310.
[8] J. G. Wilpon, C.-H. Lee, and L. R. Rabiner, "Connected digit recognition based on improved acoustic resolution," *Comput. Speech Language*, vol. 7, pp. 15–26, 1993.

Jean-Francois Mari is an assistant professor with the University of Nancy and a senior researcher with the Pattern Recognition and Artificial Intelligence Group of CRIN/INRIA, Nancy, France. His research interests relate to stochastic modeling of speech, underwater signals, and handwritten characters.

Jean-Paul Haton (F'91) heads the Pattern Recognition and Artificial Intelligence Group of CRIN/INRIA, Nancy, France. His research interests relate to artificial intelligence and man-machine communication, especially in the fields of speech recognition and understanding, spech training, and knowledge-based expert systems.

Dr. Haton is a member of AAAI, AFCET, the Acoustical Society of America, and the French Acoustical Society. He is a Fellow of the International Pattern Recognition Society (IAPR).

Abdelaziz Kriouile received the Ph.D. degree under the direction of J.-P. Haton and J.-F. Mari.

He is an assistant professor with the University of Rabat, ENSIAS, Maroc.