

AN ALGORITHM TO DETERMINE HIDDEN MARKOV MODEL TOPOLOGY

Raymond C. Vasko, Jr.

Amro El-Jaroudi

J. Robert Boston

Department of Electrical Engineering
University of Pittsburgh, Pittsburgh, PA 15261

ABSTRACT

Hidden Markov modeling (HMM) provides a probabilistic framework for modeling a time series of multivariate observations. An HMM describes the dynamic behavior of the observations in terms of movement among the states of a finite-state machine. In this paper, we present an algorithm that selects an HMM topology for a set of time series data. Our method selects a topology based on a likelihood criterion and a heuristic evaluation of complexity. The algorithm iteratively prunes state transitions from a large general HMM topology until a topology is obtained that concisely represents the dynamic structure of the data. The goal of this approach is to allow the data to reveal their own dynamic structure without external assumptions concerning the number of states or pattern of transitions.

1. INTRODUCTION

Hidden Markov models (HMMs) provide a probabilistic technique for grouping observations of a process into states. An HMM state can be interpreted as a type of behavior exhibited by the process being modeled. The HMM represents the overall process behavior in terms of movement between states and describes the inherent variations of the observations within a state [1]. For each observation, the process being modeled occupies one of the HMM states. With each observation, the HMM either moves to another state or stays in the same state, based on a set of state transition probabilities associated with the state. Thus, the state transition probability (A) distributions describe the underlying dynamic structure of the observations. The variety of the observations within a state is represented by the observation probability (B) distribution for that state, which may be either continuous or discrete. The HMMs discussed in this paper are discrete HMMs, where the observation sequence is a sequence of symbols drawn from a finite set of possible observations. With discrete HMMs, the continuous multivariate observations are mapped to a discrete set of observation symbols by a technique known as vector quantization [2].

Choosing a topology for an HMM requires *a priori* knowledge of the dynamic structure of the data to be modeled. Historically, the choice of topology for an HMM application has been determined empirically [1]. In training an HMM, the probability distributions for any topology will adapt to capture the statistical behavior of the data [3]. However, if the HMM topology has too few states or an inappropriate network of state transitions, two or more distinct types of process behavior will be represented by a single state in the HMM. This decreases the likelihood of the model for

the given data. Liu and Narayan have shown that different topologies can be equivalent in their ability to generate the same data [3]. If the data generated by two different HMMs are statistically equivalent, the simpler model is preferred, since it more efficiently represents the dynamic structure of the data.

Three main classes of topology estimators have been proposed in the literature: grammatical inference techniques [4], decomposition of large ergodic structures [5], and information-theoretic approaches [3]. The grammatical inference techniques are limited to the estimation of temporal topologies. Topology estimation by decomposition of a large ergodic HMM is not restricted to a temporal structure, but the topology estimate tends to be complex and usually not an efficient model of the process. The information-theoretic approaches are specific to the problem of estimating the order (number of states) for the HMM topology but do not address which transitions should be allowed between the states.

2. APPROACH

We have developed a method for selecting an efficient HMM topology to represent the dynamic structure of a set of time series data. The algorithm iteratively prunes state transitions (including self-transitions and possible initial states) from an initial general topology until only one state remains. The algorithm then selects a topology from the set of topologies generated over the pruning iterations.

The first step of our algorithm is to train a large general topology for the data. This topology is ergodic, with a fully-populated state transition probability (A) matrix. Each observation symbol has its own state. This topology has the most flexibility to model a set of data, since the maximum number of state transitions is available. Ten iterations of the Baum-Welch (B-W) reestimation algorithm [1] train the initial topology.

At each iteration of the algorithm, one state transition is pruned (by setting its probability to zero). To determine which transition will be removed by a pruning iteration, the algorithm trains (ten iterations of B-W) a set of T candidate topologies, where T is the number of state transitions at the input of the pruning iteration. A different state transition is removed in each candidate topology. Initial state probabilities are treated as state transition probabilities by the algorithm. That is, starting in a particular state is equivalent to a state transition from a null state into the initial state. Using the forward algorithm [1], the probability of the data given the model, $Pr(O|\lambda)$, is computed for each candidate topology. The most likely (i.e. largest $Pr(O|\lambda)$) candidate topology is then chosen as the output of the pruning iteration. Thus, a pruning iteration removes

the state transition that is least important in describing the data. When a state transition is removed, the observations that were previously using that path must use new paths, and the shapes of the model's B distributions will change. The B distributions are smoothed with a floor value of $\frac{1}{\tau}$, where τ is the total number of observations in the data. However, the algorithm does not smooth the A distributions to ensure that once a transition is pruned, it is never used.

When a pruning iteration results in the removal of a self-transition for a state, that state is eliminated entirely from the topology. A state without a self-transition cannot represent a meaningful portion of process behavior, since it is merely an intermediate step (one symbol in duration) between other states. To eliminate a state, all transitions into and out of that state are removed. In addition, state transition probabilities are redistributed to establish direct transitions between states that were previously connected via the eliminated state. Then ten iterations of B-W reestimation are performed to train the state-eliminated topology prior to the next pruning iteration. The redistribution of state transition probabilities is best explained by example. Assume that state s_2 is to be eliminated from the topology, and the probability of transition from state s_1 to state s_2 is 0.3. Also, assume that the only paths leaving state s_2 are from s_2 to state s_3 and from s_2 to state s_4 . The probability of entering state s_2 from state s_1 is uniformly distributed between the paths from s_1 to s_3 and from s_1 to s_4 . Thus, 0.15 is added to the probabilities of transition from s_1 to s_3 and from s_1 to s_4 .

The algorithm's topology estimate is selected by examining the trajectory of the $Pr(O|\lambda)$ probabilities over the pruning iterations. The trajectory of $Pr(O|\lambda)$ vs. pruning iteration is approximately flat when the removal of a state transition has not limited the topology's ability to model the data. When the removal of a state transition causes $Pr(O|\lambda)$ to decrease substantially, the topology has been pruned beyond a structure that is appropriate for modeling the data. The simplest topology reached before a substantial decrease in $Pr(O|\lambda)$ for a pruning iteration is the algorithm's estimate of the dynamic structure of the data.

3. EXAMPLE

Data consisting of 100 observation sequences with 4 possible symbols were generated by the 3-state temporal HMM shown in Figures 1 and 2. The parameters for this model are listed in Table 1. All of the observation sequences were forced to start in the same state and transition through the second state to end in the third state. Once in the third state, the probability of the end of the observation sequence was 0.1. The 100 observation sequences had a total data size of 2594 observations, with a minimum sequence length of 4 and a maximum sequence length of 91.

The pruning algorithm described in Section 2 was applied to the data. The HMM topology after each pruning iteration is shown in Figure 3, where the initial topology appears as iteration 0. This figure shows the order in which the state transitions were pruned to dissolve the initial 4-state ergodic topology to a single state. States in which an observation sequence could begin are indicated by an asterisk.

Figure 4 shows $Pr(O|\lambda)$ over the pruning iterations, normalized to compensate for the dependence of $Pr(O|\lambda)$ on data length. The algorithm reached the single-state topology in 14 pruning iterations. Since the normalized $Pr(O|\lambda)$ was decreased by pruning iteration 13, the algorithm's

topology estimate is the 3-state temporal topology obtained after pruning iteration 12. This topology (indicated by the boxed iteration number in Figure 3) matches the HMM that generated the data. The parameters of the estimated topology (shown in Table 2) also match those for the model that generated the data (Table 1).

Figure 5 illustrates how the B distributions evolve over the pruning iterations. As the A matrix becomes more sparse with the removal of each state transition, the B distributions become broader.

4. CONCLUSION

In this paper, we have described an algorithm that iteratively prunes state transitions from a large general HMM topology to estimate a topology for a set of time series data. Our topology estimate is based on both likelihood and complexity, where complexity is determined by inspection. We have presented an example of the estimation of a simple temporal HMM topology from simulated data. Our method has also shown promise in application to simulated data from ergodic topologies. Future work will involve the development of metrics that combines likelihood and complexity to select the topology estimate. In addition, metrics will be developed to quantify algorithm performance and the algorithm will be generalized to consider continuous-density HMMs. Also, the algorithm will be evaluated by simulation to determine its limitations with respect to A matrix symmetry, the smoothness and overlapping of the B distributions, vector quantization noise, and data size. The ultimate goal is to apply the algorithm to describe the dynamic structure of data from a physical process such as human sleep.

5. REFERENCES

- [1] Rabiner, L.R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, Vol. 77, No. 2, pp. 257-285, Feb. 1989.
- [2] Makhoul, J., Roucos, S., Gish, H., "Vector Quantization in Speech Coding," *Proc. IEEE*, Vol. 73, No. 11, pp. 1551-1585, Nov. 1985.
- [3] Liu, C.-C., Narayan, P., "Order Estimation and Sequential Universal Data Compression of a Hidden Markov Source by the Method of Mixtures," *IEEE Trans. on Information Theory*, Vol. 40, No. 4, pp. 1167-1180, July 1994.
- [4] Lockwood, P., Blanchet, M., "An Algorithm for the Dynamic Inference of Hidden Markov Models (DIHMM)," *IEEE Proc. ICASSP'93*, Vol. II, pp. 251-254, 1993.
- [5] Pepper, D.J., Clements, M.A., "On the Phonetic Structure of a Large Hidden Markov Model," *IEEE Proc. ICASSP'91*, pp. 465-468, 1991.

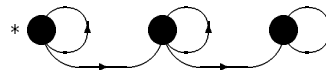


Figure 1. Actual HMM Topology

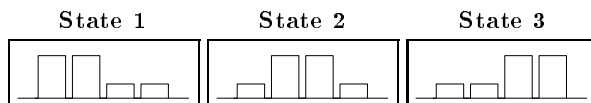


Figure 2. Actual B Distributions

A) Initial State Probability Distribution	
	$i \rightarrow$ $[\pi_i] = [1 \ 0 \ 0]$
B) State Transition Probability Distribution	
	$j \rightarrow$ $[a_{ij}] = \begin{bmatrix} 0.9 & 0.1 & 0 & 0 \\ 0 & 0.85 & 0.15 & 0 \\ 0 & 0 & 0.9 & 0 \end{bmatrix} \begin{matrix} i \\ \downarrow \end{matrix}$
C) Observation Symbol Probability Distribution	
	$k \rightarrow$ $[b_j(k)] = \begin{bmatrix} 0.375 & 0.375 & 0.125 & 0.125 \\ 0.125 & 0.375 & 0.375 & 0.125 \\ 0.125 & 0.125 & 0.375 & 0.375 \end{bmatrix} \begin{matrix} j \\ \downarrow \end{matrix}$

Table 1. Actual HMM Probability Distributions

A) Initial State Probability Distribution	
	$i \rightarrow$ $[\pi_i] = [1 \ 0 \ 0]$
B) State Transition Probability Distribution	
	$j \rightarrow$ $[a_{ij}] = \begin{bmatrix} 0.8887 & 0.1012 & 0 & 0 \\ 0 & 0.8628 & 0.1133 & 0 \\ 0 & 0 & 0.9272 & 0 \end{bmatrix} \begin{matrix} i \\ \downarrow \end{matrix}$
C) Observation Symbol Probability Distribution	
	$k \rightarrow$ $[b_j(k)] = \begin{bmatrix} 0.3510 & 0.3900 & 0.1459 & 0.1131 \\ 0.1314 & 0.3777 & 0.3606 & 0.1303 \\ 0.1061 & 0.1270 & 0.3872 & 0.3797 \end{bmatrix} \begin{matrix} j \\ \downarrow \end{matrix}$

Table 2. Estimated HMM Distributions

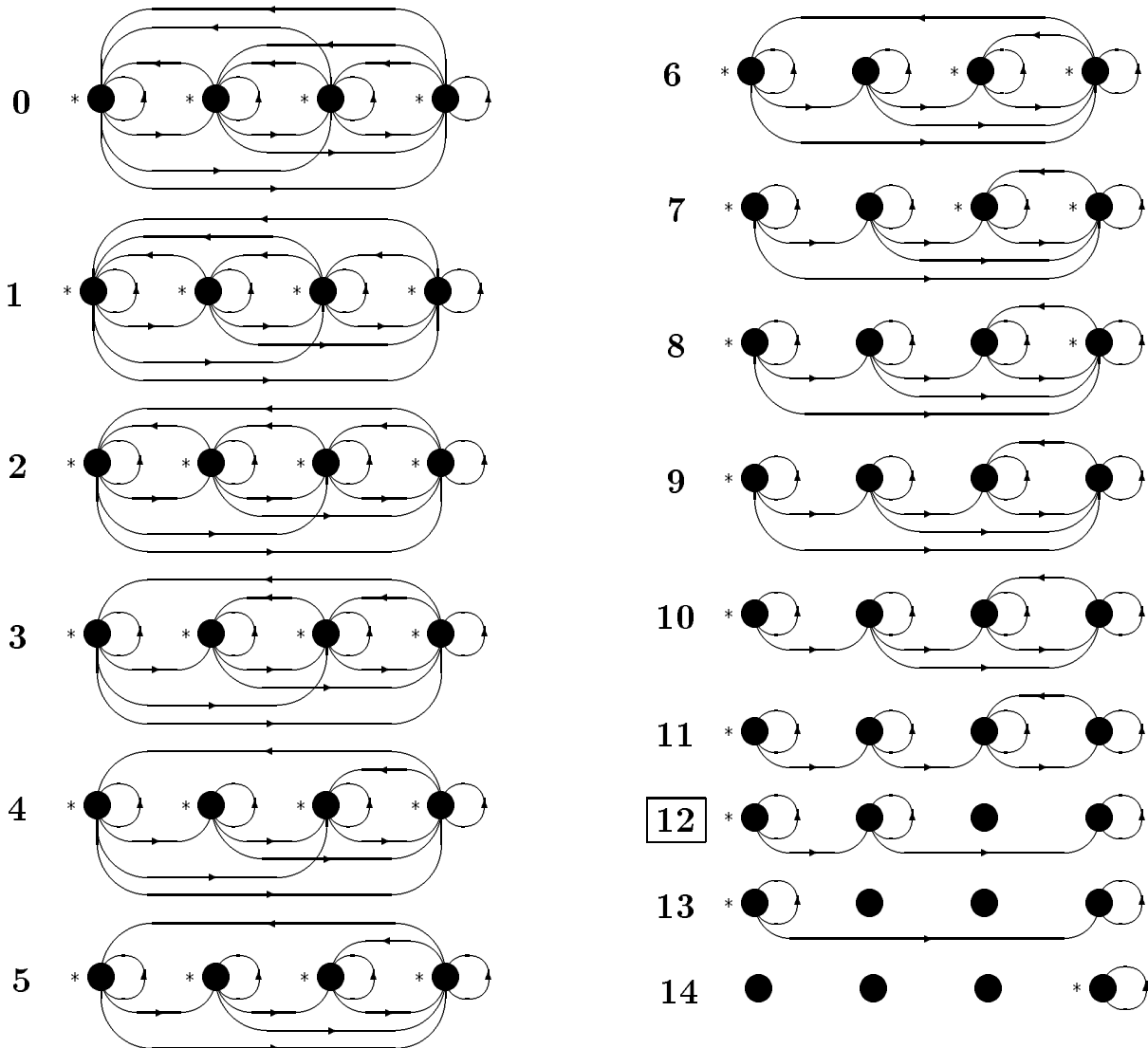


Figure 3. Cascade of HMM Topologies vs. Pruning Iteration

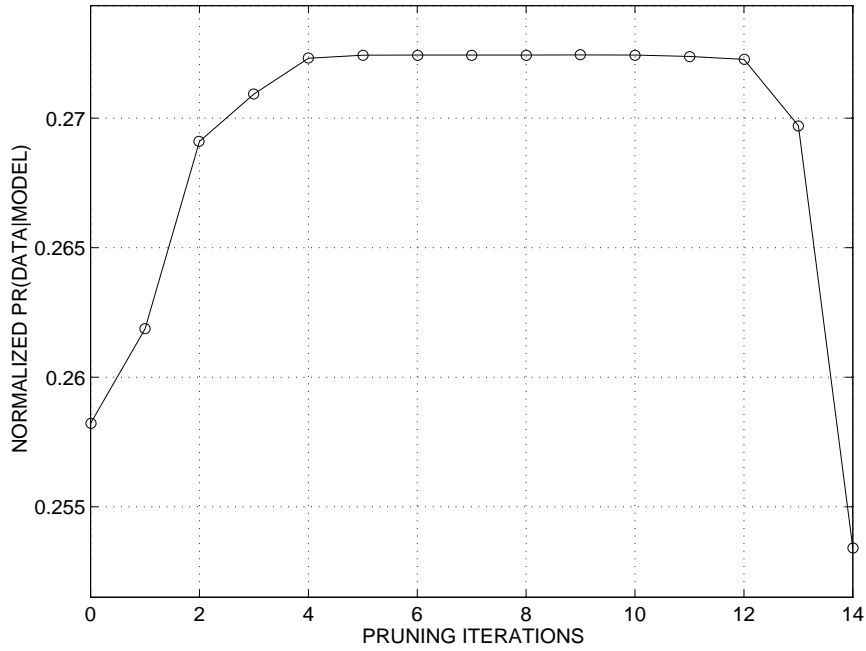


Figure 4. Plot of Normalized $Pr(O|\lambda)$ vs. Pruning Iteration

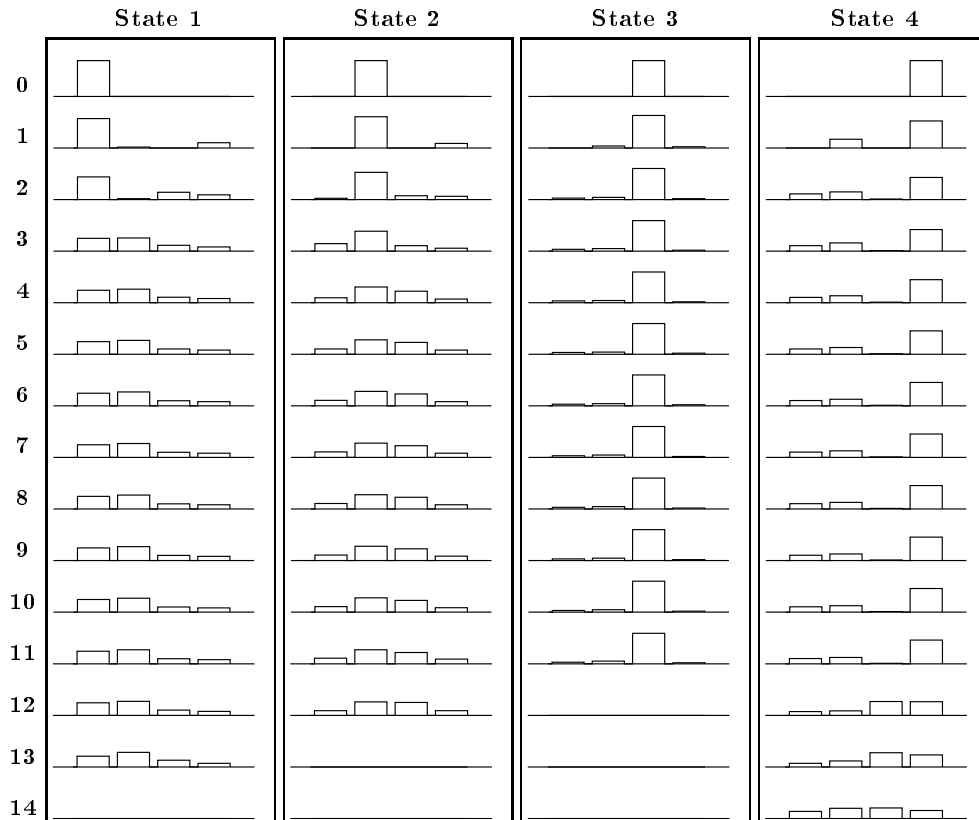


Figure 5. Cascade of B Distributions vs. Pruning Iteration