

# Finding Behavior Patterns from Temporal Data using Hidden Markov Model based Unsupervised Classification

Cen Li and Gautam Biswas

Computer Science Department

Vanderbilt University

Box 1679 Station B Nashville TN 37235

E-Mail: cenli@vuse.vanderbilt.edu, biswas@vuse.vanderbilt.edu

## Abstract

This paper describes a clustering methodology for temporal data using hidden Markov model(HMM) representation. The proposed method improves upon existing HMM based clustering methods in two ways: (i) it enables HMMs to dynamically change its model structure to obtain a better fit model for data during clustering process, and (ii) it provides objective criterion function to automatically select the optimal clustering partition. The algorithm is presented in terms of four nested levels of searches: (i) the search for the optimal number of clusters in a partition, (ii) the search for the optimal partition structure, (iii) the search for the optimal HMM structure for each cluster, and (iv) the search for the optimal parameter values for each HMM. Preliminary experiments with artificially generated data demonstrate the effectiveness of the proposed methodology.

## Introduction

Unsupervised classification, or clustering, assumes data is not labeled with class information. The goal is to create structure for data by objectively partitioning data into homogeneous groups where the within group object similarity and the between group object dissimilarity are optimized. Data categorization is achieved by analyzing and interpreting feature descriptions associated with each group. The technique has been used extensively by researchers in discovering structures from databases where domain knowledge is not available or incomplete(Cheeseman & Stutz 1996)(Biswas, Weinberg, & Li 1995).

In the past, the focus of clustering analysis has been on data described with static features(Cheeseman & Stutz 1996)(Biswas, Weinberg, & Li 1995)(Fisher 1987)(Wallace & Dowe 1994), i.e., values of the features do not change, or the changes are negligible, during observation period. Examples of static features include customer's age, education level and salary, or patient's age, gender, weight, and previous diseases upon hos-

pital admission. In real world, most systems are dynamic which often are best described by temporal features, whose values change significantly during observation period. Examples of temporal features include daily ATM transactions and account balances of bank customers, and frequent recordings of blood pressure, temperature and respiratory rate of patients under intensive hospital care. Clustering data described with static features identifies patterns from data in terms of time-invariant characteristics. Clustering data described with temporal features aimed at profiling behavior patterns for dynamic systems through data partitioning and cluster interpretation. Clustering temporal data is inherently more complex than clustering static data. First of all, the dimensionality of the data is significantly larger in temporal case. When data objects are characterized using static features, only one value is present for each feature. In temporal feature case, each feature is associated with a sequence of values. Also, the complexity of cluster definition(modeling) and interpretation increases by orders of magnitude with dynamic data(Li 1998).

We choose hidden Markov model representation for our profiling problem. A HMM is a non-deterministic stochastic Finite State Automata(FSA). The main characteristic that differentiates HMM from Markov chains is that, with HMM, the state sequence is not directly observed. Only the generated value sequence, a probabilistic function of the underlying value sequence, is observable. From a dynamic system's viewpoint, the value sequences are considered the manifestation of the dynamic, stochastic behavior of the system. The basic structure of a HMM consists of a connected set of hidden states. HMM models can be described by the following three sets of probabilities: (i) the initial state probabilities,  $\pi$ , which defines the probability of each state being the starting state for any value sequence, (ii) the transition probability matrix,  $A$ , which defines the probability of going from one state to another, and (iii) the emission probability matrix,  $B$ , which defines the probability of generating a value at any given state(Rabiner 1989). We are interested in building HMMs for continuous temporal sequences, where the temporal

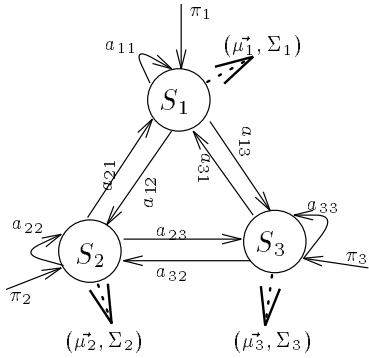


Figure 1: An Example 3-State HMM

feature values are continuous rather than symbolic. In our case, the emission probability density function (*pdf*) within each state is defined by a multivariate Gaussian distribution characterized by its mean vector,  $B_{\bar{\mu}}$ , and co-variance matrix,  $B_{\Sigma}$ . An example of a first order continuous density HMM with 3 states is shown in Figure 1. The  $\pi_i$ s are the initial state probabilities for state  $i$ . The  $a_{ij}$ s are the transition probabilities from state  $i$  to state  $j$  and the  $(\bar{\mu}_i, \Sigma_i)$ s define the *pdfs* for emission probabilities for state  $i$ .

There are a number of advantages in using the HMM representation for our problem:

- There are direct links between the HMM states and real world situations for the problem under consideration. The hidden states of a HMM can be used to effectively model the set of potentially valid states of a dynamic process. While the exact sequence of stages going through by a dynamic system may not be observed, it can be estimated based on observable behavior of the systems.
- HMMs represent a well-defined probabilistic model. The parameters of a HMM can be determined in a precise, well-defined manner, using methods such as maximal likelihood estimates or maximal mutual information criterion.
- HMMs are graphical models of underlying dynamic processes that govern system behavior. Graphical models may aid the interpretation task.

## Proposed HMM Clustering Methodology

Clustering using HMMs was first mentioned by Rabiner *et al.* (Rabiner *et al.* 1989) for speech recognition problems. The idea has been further explored by other researchers including Lee (Lee 1990), Dermatas and Kokkinakis (Dermatas & Kokkinakis 1996), Lee (Lee 1990), Kosaka *et al.* (Kosaka, Masunaga, & Kuraoka 1995), and Smyth (Smyth 1997). Two main problems that have been identified in these works are: (i) no objective criterion measure is used for determining the

optimal size of the clustering partition, and (ii) uniform, pre-specified HMM structure is used for different clusters of each partition. This paper describes a HMM clustering methodology that tries to remedy these two problems by developing an objective partition criterion measure based on model mutual information, and by developing an explicit HMM model refinement procedure that dynamically modify HMM structures during clustering process.

The proposed HMM clustering method can be summarized in terms of four levels of nested searches. From the outer most to the inner most level, the four searches are: the search for

1. the optimal number of clusters in a partition,
2. the optimal structure for a given partition,
3. the optimal HMM structure for each cluster, and
4. the optimal HMM parameters for each cluster.

Starting from the inner most level of search, each of these four search steps are described in more detail next.

### Search Level 4: HMM Parameter Reestimation

This step tries to find the maximal likelihood parameters for the HMM of a fixed size. The well known Baum-Welch parameter reestimation procedure (Baum *et al.* 1970) is used for this purpose. The *Baum-Welch* procedure is a variation of the more general EM algorithm (Dempster, Laird, & Rubin 1977), which iterates between two steps: (i) the expectation step (E-step), and (ii) the maximization step (M-step). The E-step assumes the current parameters of the model and computes the expected values of a necessary statistics. The M-step uses these statistics to update the model parameters so as to maximize the expected likelihood of the parameters (Ghahramani & Jordan 1997). The procedure is implemented using the *forward-backward* computations.

### Search Level 3: The Optimal HMM Structure

This step attempts to replace an existing model for a group of objects by a more accurate and refined HMM model. Solcke and Omohundro (Stolcke & Omohundro 1994) described a technique for inducing the structure of HMMs from data based on a general “model merging” strategy (Omohundro 1992). Takami and Sagayama (Takami & Sagayama 1992) proposed the Successive State Splitting (SSS) algorithm to model context-dependent phonetic variations. Ostendorf and Singer (Ostendorf & Singer 1997) further expanded the basic SSS algorithm by choosing the node and the candidate split at the same time based on the likelihood gains. Casacuberta *et al.* (Casacuberta, Vidal, & Mas 1990) proposed to derive the structure of HMM through

error correcting grammatical inference techniques.

Our HMM refinement procedure combines ideas from the past works. We start with an initial model configuration and incrementally grow or shrink the model through HMM state splitting and merging operations for choosing the right size model. The goal is to obtain a model that can better account for the data, i.e., having a higher model posterior probability. For both merge and split operations, we assume the Viterbi path does not change after each operation, that is for the split operation, the observations that were in state  $s$  will reside in either one of the two new states,  $q_0$  or  $q_1$ . The same is true for the merge operation. This assumption can greatly simplify the parameter estimation process for the new states. The choice of state(s) to apply the split(merge) operation is dependent upon the state emission probabilities. For the split operation, the state that has the highest variances is split. For the merge operation, the two states that have the closest mean vector are considered for merging. Next we describe the criterion measure used to perform heuristic model selection during HMM refinement procedure.

**Bayesian Information Criterion(BIC) for HMM Model Selection** Li and Biswas(Li & Biswas 1999) proposed one possible HMM model selection criterion, the Posterior Probability of HMM(PPM), which is developed based on Bayesian model merging criterion in (Stolcke & Omohundro 1994). One problem with the PPM criterion is that it depends heavily on the base values for the exponential distributions used to compute prior probabilities of global model structures of HMMs.

Here, we present an alternative HMM model selection scheme. From Bayes theorem, given data,  $X$ , and a model,  $\lambda$ , trained from  $X$ , the posterior probability of the model,  $P(\lambda|X)$ , is given by:

$$P(\lambda|X) = \frac{P(\lambda)P(X|\lambda)}{P(X)},$$

where  $P(X)$  and  $P(\lambda)$  are prior probabilities of the data and the model respectively, and  $P(X|\lambda)$  is the marginal likelihood of data. Since the prior probability of data remains unchanged for different models, for model comparison purpose, we have  $P(\lambda|X) \propto P(\lambda)P(X|\lambda)$ . By assuming uniform prior probability for different models,  $P(\lambda|X) \propto P(X|\lambda)$ . That is, the posterior probability of a model is directly proportional to the marginal likelihood. Therefore, the goal is to select the model that gives the highest marginal likelihood.

Computing marginal likelihood for complex models has been an active research area (Kass & Raftery 1995) (Chichering & Heckerman 1997) (Cooper & Herskovits 1992) (Chib 1995). Approaches include Monte-Carlo methods, i.e., Gibbs sampling methods (Chib 1995) (G. & I. 1992), and various approximation meth-

ods, i.e., the Laplace approximation (Kass & Raftery 1995) and approximation based on Bayesian information criterion (Chichering & Heckerman 1997). It has been well documented that although the Monte-Carlo methods are very accurate, they are computationally inefficient especially for large databases. It is also shown that under certain regularity conditions, Laplace approximation can be quite accurate, but its computation can be expensive, especially for its component Hessian matrix computation.

A widely used and very efficient approximation method for marginal likelihood is Bayesian information criterion where, in log form, marginal likelihood of a model given data is computed as:

$$\log P(\lambda|X) = \log P(X|\lambda, \theta_\lambda) - \frac{d}{2} \log N,$$

where  $\theta_\lambda$  is Maximum Likelihood(ML) configuration of the model,  $d$  is the dimensionality of the model parameter space and  $N$  is the number of cases in data. The first term in BIC computation,  $\log P(X|\lambda, \theta_\lambda)$ , is the likelihood term which tends to promote larger and more detailed models of data, whereas the second term,  $-\frac{d}{2} \log N$ , is the penalty term which favors smaller model having less parameters. BIC selects the most appropriate model for data by balancing these two terms. Chichering and Heckerman discussed the close similarities between BIC measure and Minimal Description Length(MDL) principle(Chichering & Heckerman 1997).

We use BIC as our HMM model selection criterion. Here, we use an example to illustrate how BIC helps to select the correct model structure from data. An artificial data set of 100 data objects is generated from a predefined five-state HMM. Each data object is described using two temporal features. The length of temporal sequences of each feature is 50. Figure 2 shows values of the likelihood term, the penalty term, and the BIC measure, when the same data set is modeled using HMMs of sizes ranging from 2 to 10. The dotted line shows the likelihoods of data modeled using HMMs of different sizes. The dashed lines shows the penalties incurred for each model. And the solid line shows the BIC measures as a combination of the above two terms. We observe, as the size of the model increases, the model likelihood also increases, but the model penalty term decreases. BIC correctly identifies the true model, the 5-state model, for this data.

## Search Level 2: The Optimal Partition Structure

The two most commonly used distance measures in the context of the HMM representation is the sequence-to-model likelihood measure (Rabiner 1989) and the symmetrized distance measure between pairwise models (Juang & Rabiner 1985). We choose the sequence-to-model likelihood distance measure for our HMM



rect model structure is known and fixed throughout the clustering process in this example. To generate data with  $K$  clusters, first we manually create  $K$  HMMs. From each of these  $K$  HMMs, we generate  $N_k$  objects, each described with  $M$  temporal sequences. The length of each temporal sequence is  $L$ . The total data points for such a data set is  $K \cdot N_k \cdot M \cdot L$ . In these experiments, we choose  $K = 4$ ,  $N_k = 30$ ,  $M = 2$ , and  $L = 100$ . The HMM for each cluster has 5 states.

First, the PMI criterion measure was not incorporated in the binary clustering tree building process. The branches of the tree is terminated either because there are too few objects in the node, or because the object redistribution process in a node ends with one cluster partition. The *full* binary clustering tree, as well as the PMI scores for intermediate and final partitions are computed and shown in Figure 3(a). The PMI scores to the right of the tree indicate the quality of the current partition, which includes all nodes at the frontier of the current tree. For example, the PMI score for the partition having clusters  $C_4$  and  $C_{123}$  is 0.0, and PMI score for the partition having clusters  $C_4$ ,  $\frac{4}{30}C_2$ ,  $\frac{26}{30}C_2$ , and  $C_{13}$  is  $-1.75 * 10^2$ . The result of this clustering process is a 7-cluster partition, with six fragmented clusters, i.e., cluster  $C_2$  is fragmented into  $\frac{4}{30}C_2$  and  $\frac{26}{30}C_2$ , cluster  $C_3$  is fragmented into  $\frac{1}{30}C_3$ ,  $\frac{29}{30}C_3$ , and cluster  $C_1$  is fragmented into  $\frac{4}{30}C_1$  and  $\frac{26}{30}C_1$ . Figure 3(b) shows the binary HMM clustering tree where PMI criterion measure is used for determining branch terminations. The dotted lines cut off branches of the search tree where the split of the parent cluster results in a decrease in the PMI score. This clustering process re-discovers the correct 4-cluster partition.

## Experiment

In this experiment, we combine the four search steps in performing unsupervised classification on artificial data generated from models of different sizes. First, we created three HMMs: one with three states, one with four states, and one with five states. Based on each model, 50 data objects are created, each described by two temporal features. The sequence length for each temporal feature is 50. Figure 4 shows six example data objects from this data set. The dotted lines and the solid lines represent values of the two temporal features for each object. It is observed that, from the feature values, it is quite difficult to differentiate which objects are generated from which model. In fact, objects (a) and (f) are generated from the three-state HMM, objects (b) and (e) are generated from the four-state HMM, and objects (c) and (d) are generated from the five-state HMM. Detailed parameters of these three models are given in the appendix.

Given this data, our method successfully uncovers the correct clustering partition size, i.e., 3 clusters in the partition, and individual data object is assigned to

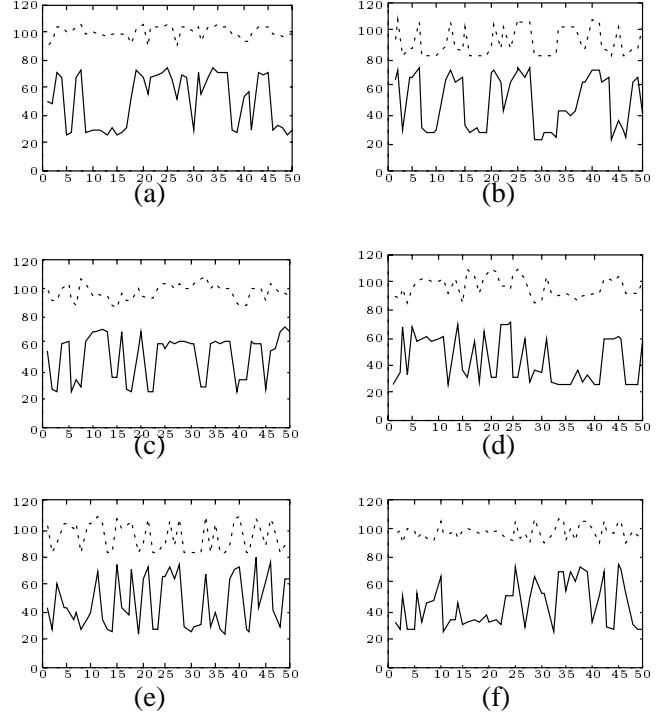


Figure 4: Compare Data Objects Generated from Different Models

the correct cluster, i.e., the cluster whose derived model corresponds to the object’s generative model. Furthermore, for each cluster, our method accurately reconstructed the HMM with the correct model size and near perfect model parameter values.

## Conclusion

We presented a temporal data clustering methodology based on HMM representation. HMMs have been used in speech recognition problems to model human pronunciations. Since the main objective in that study is recognition, it is not essential whether the true model structure is uncovered. A fixed size model structure can be used throughout data analyses, as long as the model structure is adequate in *differentiating* objects coming from different underlying models. On the other hand, in our case, HMMs are used to profile temporal behaviors of dynamic systems. Our ultimate objective is to characterize behavior patterns of dynamic systems by interpreting the HMMs induced from temporal data. Therefore, it is extremely important that the derived models are as close to the underlying models as possible. To facilitate this, we introduced a dynamic HMM refinement procedure to the clustering process and employed an objective measure, BIC, for model selection purposes. Furthermore, we have developed the PMI criterion measure for selecting the optimal partition size. This allows an objective and automatic clustering process which can be very useful in many discovery tasks.

Our next step is to apply this method to real world problems. The application domain we are currently studying is about pediatric patients having Respiratory Distress Syndrome(RDS) and undergoing intensive hospital care. The goal of this application is to identify patient response patterns from temporal data recorded in the form of vital signs measured frequently throughout a patient's stay at the hospital.

## Appendix

Parameters of the three HMMs used in the final experimentation are given here.

### Three-state HMM

$$\begin{aligned} \pi &= (0.4, 0.4, 0.2) \\ A &= \begin{pmatrix} 0.5 & 0.25 & 0.15 \\ 0.35 & 0.4 & 0.25 \\ 0.25 & 0.3 & 0.55 \end{pmatrix} \\ B_{\bar{\mu}} &= \left( \begin{array}{c} \begin{bmatrix} 30 \\ 98.7 \end{bmatrix} \\ \begin{bmatrix} 55 \\ 92 \end{bmatrix} \\ \begin{bmatrix} 70 \\ 105 \end{bmatrix} \end{array} \right) \\ B_{\Sigma} &= \left( \begin{array}{c} \begin{bmatrix} 3 \\ 0.8 \end{bmatrix} \\ \begin{bmatrix} 3 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 2 \\ 1.2 \end{bmatrix} \end{array} \right) \end{aligned}$$

### Four-state HMM

$$\begin{aligned} \pi &= (0.2, 0.25, 0.3, 0.25) \\ A &= \begin{pmatrix} 0.3 & 0.15 & 0.25 & 0.3 \\ 0.15 & 0.3 & 0.3 & 0.25 \\ 0.05 & 0.05 & 0.65 & 0.25 \\ 0.25 & 0.05 & 0.65 & 0.25 \end{pmatrix} \\ B_{\bar{\mu}} &= \left( \begin{array}{c} \begin{bmatrix} 40 \\ 102 \end{bmatrix} \\ \begin{bmatrix} 72 \\ 107 \end{bmatrix} \\ \begin{bmatrix} 28 \\ 84 \end{bmatrix} \\ \begin{bmatrix} 65 \\ 89 \end{bmatrix} \end{array} \right) \\ B_{\Sigma} &= \left( \begin{array}{c} \begin{bmatrix} 2.5 \\ 0.3 \end{bmatrix} \\ \begin{bmatrix} 3 \\ 0.8 \end{bmatrix} \\ \begin{bmatrix} 2 \\ 0.5 \end{bmatrix} \\ \begin{bmatrix} 2.5 \\ 0.8 \end{bmatrix} \end{array} \right) \end{aligned}$$

### Five-state HMM

$$\begin{aligned} \pi &= (0.2, 0.1, 0.25, 0.3, 0.15) \\ A &= \begin{pmatrix} 0.3 & 0.2 & 0.1 & 0.2 & 0.2 \\ 0.3 & 0.4 & 0.05 & 0.15 & 0.1 \\ 0.1 & 0.05 & 0.45 & 0.2 & 0.2 \\ 0.15 & 0.05 & 0.05 & 0.55 & 0.2 \\ 0.05 & 0.1 & 0.05 & 0.3 & 0.5 \end{pmatrix} \\ B_{\bar{\mu}} &= \left( \begin{array}{c} \begin{bmatrix} 36 \\ 88 \end{bmatrix} \\ \begin{bmatrix} 69 \\ 97 \end{bmatrix} \\ \begin{bmatrix} 30 \\ 107 \end{bmatrix} \\ \begin{bmatrix} 26 \\ 92 \end{bmatrix} \\ \begin{bmatrix} 60 \\ 102 \end{bmatrix} \end{array} \right) \\ B_{\Sigma} &= \left( \begin{array}{c} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \\ \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} \end{array} \right) \end{aligned}$$

## References

Bahl, L. R.; Brown, P. F.; De Souza, P. V.; and Mercer, R. L. 1986. Maximum mutual information estimation of hidden markov model parameters. In *Proceedings of the IEEE-IECEJ-AS International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 49–52.

Baum, L. E.; Petrie, T.; Soules, G.; and Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics* 4(1):164–171.

Biswas, G.; Weinberg, J.; and Li, C. 1995. Iterate: A conceptual clustering method for knowledge discovery in databases. In Braunschweig, B., and Day, R., eds., *Artificial Intelligence in Petroleum Industry: Symbolic and Computational Applications*. Teditons Technip.

Casacuberta, F.; Vidal, E.; and Mas, B. 1990. Learning the structure of hmm's through grammatical inference techniques. In *Proceedings of the International Conference on Acoustic, Speech, and Signal Processing*, 717–720.

Cheeseman, P., and Stutz, J. 1996. Bayesian classification(autoclass): Theory and results. In Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining*. AAAI-MIT press. chapter 6, 153–180.

Chib, S. 1995. Marginal likelihood from the gibbs sampling. *Journal of the American Statistical Association* 1313–1321.

Chichering, D. M., and Heckerman, D. 1997. Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine Learning* 29:181–212.

Cooper, G. F., and Herskovits, E. 1992. A bayesian method for the induction of probabilistic network from data. *Machine Learning* 9:309–347.

Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of Royal Statistical Society Series B(methodological)* 39:1–38.

Dermatas, E., and Kokkinakis, G. 1996. Algorithm for clustering continuous density hmm by recognition error. *IEEE Transactions on Speech and Audio Processing* 4(3):231–234.

Fisher, D. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine Learning* 2:139–172.

G., C., and I., G. E. 1992. Explaining the gibbs sampler. *The American Statistician* 46(3):167–174.

Ghahramani, Z., and Jordan, M. I. 1997. Factorial hidden markov models. *Machine Learning* 29:245–273.

Juang, B. H., and Rabiner, L. R. 1985. A probabilistic distance measure for hidden markov models. *AT&T Technical Journal* 64(2):391–408.

Kass, R. E., and Raftery, A. E. 1995. Bayes factor. *Journal of the American Statistical Association* 773–795.

Kosaka, T.; Masunaga, S.; and Kuraoka, M. 1995. Speaker-independent phone modeling based on speaker-dependent hmm's composition and clustering. In *Proceedings of the ICASSP' 95*, 441–444.

- Lee, K. F. 1990. Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38(4):599-609.
- Li, C., and Biswas, G. 1999. Clustering sequence data using hidden markov model representation. In *Proceedings of SPIE99 conference on Data Mining and Knowledge Discovery*.
- Li, C. 1998. Unsupervised classification on temporal data. Technical Report VU-CS-TR-98-04, Vanderbilt University, Box 1679 B, Department of Computer Science, Vanderbilt Univ. Nashville, TN 37235.
- Omohundro, S. M. 1992. Best-first model merging for dynamic learning and recognition. *Advances in Neural Information Processing Systems* 4:958-965.
- Ostendorf, M., and Singer, H. 1997. Hmm topology design using maximum likelihood successive state splitting. *Computer Speech and Language* 11:17-41.
- Rabiner, L. R.; Lee, C. H.; Juang, B. H.; and Wilpon, J. G. 1989. Hmm clustering for connected word recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*.
- Rabiner, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257-285.
- Smyth, P. 1997. Clustering sequences with hidden markov models. *Advances in Neural Information Processing*.
- Stolcke, A., and Omohundro, S. M. 1994. Best-first model merging for hidden markov model induction. Technical Report TR-94-003, International Computer Science Institute, 1947 Center St., Suite 600, Berkeley, CA 94704-1198.
- Takami, J., and Sagayama, S. 1992. A successive state splitting algorithm for efficient allophone modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 1*, 573-576.
- Wallace, C. S., and Dowe, D. L. 1994. Intrinsic classification by mml - the snob program. In *Proceedings of the Seventh Australian Joint Conference on Artificial Intelligence*, 37-44. World Scientific.