
A Bayesian Approach to Temporal Data Clustering using Hidden Markov Models

Cen Li
Gautam Biswas

CENLI@VUSE.VANDERBILT.EDU
BISWAS@VUSE.VANDERBILT.EDU

Department of EECS, Box 1679 Station B, Vanderbilt University, Nashville, TN 37235 USA

Abstract

This paper presents clustering techniques that partition temporal data into homogeneous groups, and constructs state based profiles for each group in the hidden Markov model (HMM) framework. We propose a Bayesian HMM clustering methodology that improves upon existing HMM clustering by incorporating HMM model size selection into clustering control structure to derive better cluster models and partitions. Experimental results indicate the effectiveness of our methodology.

1. Introduction

Unsupervised classification, or clustering, derives structure from data by using objective criteria to partition the data into homogeneous groups so that the within group object similarity and the between group object dissimilarity are optimized simultaneously. Categorization and interpretation of structure are achieved by analyzing the models constructed in terms of the feature value distributions within each group. In the past, cluster analysis techniques have focused on data described by static features. In many real applications, the dynamic characteristics, i.e., how a system interacts with the environment and evolves over time, are of interest. Such behavior or characteristic of these systems is best described by temporal features whose values change significantly during the observation period. Our goal for temporal data clustering is to construct profiles of dynamic processes by constructing and analyzing well defined, parsimonious models of data.

Clustering temporal data is inherently more complex than clustering static data because: (i) the dimensionality of data is significantly larger in the temporal case, and (ii) the complexity of the models that describe the cluster structures, and their interpretation

is much more complex. We assume that the temporal data sequences that define the dynamic characteristics of the phenomenon under study satisfy the Markov property, and the data generation may be viewed as a probabilistic walk through a fixed set of states. When state definitions directly correspond to feature values, a Markov chain model representation of the data may be appropriate (Sebastiani et al., 1999). When the state definitions are not directly observable, or it is not feasible to define states by exhaustive enumeration of feature values, i.e., data is described with multiple continuous valued temporal features, the hidden Markov model (HMM) is appropriate. In this paper, we focus on temporal data clustering using the HMM representation.

Our ultimate goal is to develop through the extracted HMM models, an accurate and explainable representation of the system dynamics. It is important for our clustering system to determine the best number of clusters to partition the data, and the best model structure, i.e., the number of states in a model, to characterize the dynamics of the homogeneous data within each cluster. We approach these tasks by (i) developing an explicit HMM model size selection procedure that dynamically modifies the size of the HMMs during the clustering process, and (ii) casting the HMM model size selection and partition selection problems in terms of a Bayesian model selection problem.

2. Modeling with HMMs

A HMM is a non-deterministic stochastic Finite State Automata. The basic structure of a HMM consists of a connected set of states, $S = (S_1, S_2, \dots, S_n)$. We use first order HMMs, where the state of a system at a particular time t is only dependent on the state of the system at the immediate previous time point, i.e., $P(S_t | S_{t-1}, S_{t-2}, \dots, S_1) = P(S_t | S_{t-1})$. In addition, we assume all the temporal feature values are continuous, therefore, we use the continuous density HMM

(CDHMM) representation where all temporal features have continuous values. A CDHMM of n states for data having m temporal features can be characterized¹ in terms of three sets of probabilities (Rabiner, 1989): the initial state probabilities, the transition probability, and the emission probabilities. The initial state probabilities, $\vec{\pi}$ of size n , defines the probability of any of the given states being the initial state of the given sequence. The transition probability matrix, A of size $n \times n$, defines the probability of transition from state i at time t , to state j at the next time step. And the emission probability matrix, B of size $n \times m$, defines the probability of generating feature values at any given state. For CDHMM, the emission probability density function of each state is defined by a multivariate Gaussian distribution.

There are a number of advantages to using the HMM representation for the data profiling. First, the hidden states of a HMM can be used to effectively model the set of potentially valid states in a dynamic process. While the complete set of states and the exact sequence of states a system goes through may not be observable, it can be estimated by studying the observed behaviors of the system. Second, HMMs represent a well-defined probabilistic model. The parameters of a HMM can be determined in a well-defined manner. Furthermore, HMMs provide a graphical state based representation of the underlying dynamic processes that govern system behavior. Graphical models may aid the cluster analysis and model interpretation tasks.

3. The HMM Clustering Problem

HMM clustering incorporating HMM model size selection can be described in terms of four nested search steps:

- Step 1: the number of clusters in a partition;
- Step 2: the object distribution to clusters in a given partition size;
- Step 3: the HMM model sizes for individual clusters in the partition; and
- Step 4: the HMM parameter configuration for the individual clusters.

One primary limitation of the earlier work on HMM clustering (Rabiner et al., 1989), (Dermatas & Kokkinakis, 1996), (Kosaka et al., 1995), (Smyth, 1997) is that for search step 1, no objective criterion measure is used to automatically select the cluster partition based on data. A pre-determined threshold value on

¹We assume the continuous features are sampled at a pre-defined rate, and the temporal feature values are defined as a sequence of values.

data likelihood, or a post-clustering Monte-Carlo simulation, is used instead. Another limitation is that they assume a pre-specified and uniform HMM size for all models in the intermediate and final clusters in a partition. Therefore, search step 3 does not exist in those systems. There is a separate line of work that derives the HMM model structure from homogeneous data (Stolcke & Omohundro, 1994) (Ostendorf & Singer, 1997), i.e., no clustering is involved. The main difficulty of directly applying their methods in our HMM clustering problem is that these methods either do not scale to continuous valued temporal data, or they do not have an objective way of comparing quality of models of different sizes.

Once a model size (i.e., the number of states in the HMM model) is selected, step 4 is invoked to select model parameters that optimize a chosen criterion. A well known Maximum Likelihood (ML) parameter estimation method, the *Baum-Welch* procedure (Rabiner, 1989), is a variation of the more general EM algorithm (Dempster et al., 1977). It iterates between an expectation step (E-step) and a maximization step (M-step). The E-step assumes the current parameter configuration of the model and computes the expected values of necessary statistics. The M-step uses these statistics to update the model parameters so as to maximize the expected likelihood of the parameters. The iterative process continues until the parameter configuration converges.²

4. Bayesian Clustering Methodology

4.1 Bayesian Clustering

In model-based clustering, it is assumed that data is generated by a mixture of underlying probability distributions. The mixture model, M , is represented by K component models and a hidden, independent discrete variable C , where each value i of C represents a component cluster, modeled by λ_i . Given observations $X = (x_1, \dots, x_N)$, let $f_k(x_i|\theta_k, \lambda_k)$ be the density of an observation x_i from the k th component model, λ_k , where θ_k is the corresponding parameters of the model. The likelihood of the mixture model given data is expressed as: $P(X|\theta_1, \dots, \theta_K, P_1, \dots, P_K) = \prod_{i=1}^N \sum_{k=1}^K P_k \cdot f_k(x_i|\theta_k, \lambda_k)$, where P_k is the probability that an observation belongs to the k th component ($P_k \geq 0, \sum_{k=1}^K P_k = 1$). Bayesian clustering casts the model-based clustering problem into the Bayesian model selection problem. Given partitions with different component clusters, the goal is to se-

²A HMM converges when the log data likelihood given two consecutive model configurations differ less than 10^{-6} .

lect the best overall model, M , that has the highest *posterior probability*, $P(M|X)$.

4.2 Model Selection Criterion

From Bayes theorem, the posterior probability of the model, $P(M|X)$, is given by: $P(M|X) = \frac{P(M)P(X|M)}{P(X)}$, where $P(X)$ and $P(M)$ are prior probabilities of the data and the model respectively, and $P(X|M)$ is the marginal likelihood of the data. For the purpose of comparing alternate models, we have $P(M|X) \propto P(M)P(X|M)$. Assuming none of the models considered is favored a priori, $P(M|X) \propto P(X|M)$. That is, the posterior probability of a model is directly proportional to the marginal likelihood. Therefore, the goal is to select the mixture model that gives the highest marginal likelihood.

Given the parameter configuration, θ , of a model M , the marginal likelihood of the data is computed as $P(X|M) = \int_{\theta} P(X|\theta, M)P(\theta|M)d\theta$. When parameters involved are continuous valued, as in the case of CDHMM, the integration computation often becomes too complex to obtain a closed form solution. Common approaches include Monte-Carlo methods, i.e., Gibbs sampling methods (Chib, 1995), and various approximation methods, such as the Laplace approximation (Kass & Raftery, 1995). It has been well documented that although the Monte-Carlo methods and the Laplace approximation are quite accurate, they are computationally intense. Next, we look at two efficient approximation methods: the Bayesian information criterion (BIC) (Heckerman et al., 1995), and the Cheeseman-Stutz (CS) approximation (Cheeseman & Stutz, 1996).

4.2.1 BAYESIAN INFORMATION CRITERION

BIC is derived from the Laplace approximation (Heckerman et al., 1995):

$$\log P(M|X) \approx \log P(X|M, \hat{\theta}) - \frac{d}{2} \log N,$$

where d is the number of parameters in the model, N is the number of data objects, and $\hat{\theta}$ is the ML parameter configuration of model M . $\log P(X|M, \hat{\theta})$, the data likelihood, 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 models with less parameters. BIC selects the best model for the data by balancing these two terms.

4.2.2 CHEESEMAN-STUTZ APPROXIMATION

Cheeseman and Stutz (Cheeseman & Stutz, 1996) first proposed the CS approximation method for their

Bayesian clustering system, AUTOCLASS. $P(X|M) = P(X'|M) \frac{P(X|M)}{P(X'|M)}$, where X' represents complete data, i.e., data with known cluster labels. The first term is the marginal likelihood of the complete data. The exact integration is approximated by a summation over a set of local maximum parameter configurations, θ_s : $P(X'|M) \approx \sum_{\theta \in \theta_s} P(\theta|M)P(X'|\theta, M)$. To further reduce the computation burden, in our work, we have taken this approximation further by using a single maximum likelihood configuration, $\hat{\theta}$, $\hat{\theta} \in \theta_s$, to approximate the summation, i.e., $P(X'|M) \approx P(\hat{\theta}|M)P(X'|\hat{\theta}, M)$.

The second term in the CS approximation is a gross adjustment term. Both its numerator and denominator are expanded using the BIC measure. Ignoring differences between the penalty terms in the numerator and the denominator, we obtain:

$$\log P(X|M) \approx \log P(\hat{\theta}|M) + \log P(X|\hat{\theta}, M),$$

where X is the incomplete data and $P(\hat{\theta}|M)$ is the prior probability of the ML model parameter values.

5. Bayesian HMM Clustering

Recently, graphical models have been incorporated into the Bayesian clustering framework. Components of a Bayesian mixture model, which typically are modeled as multivariate normal distributions (Cheeseman & Stutz, 1996) are replaced with graphical models such as Bayesian networks (Thiesson et al., 1998) and Markov chain models (Sebastiani et al., 1999). We have adapted Bayesian clustering to CDHMM clustering, so that: (i) components of a Bayesian mixture model are represented by CDHMMs, and (ii) $f_k(X_i|\theta_k, \lambda_k)$ in data likelihood computation computes the likelihood of a multi-feature temporal sequence given a CDHMM.

First, we describe how the general Bayesian model selection criterion is adapted for the HMM model size selection and the cluster partition selection problems. Then we describe how the characteristics of these criterion functions are used to design our heuristic clustering search control structure.

5.1 Criterion Functions

5.1.1 CRITERION FOR HMM SIZE SELECTION

The HMM model size selection process picks the HMM with the number of states that best describe the data. We use Bayesian model selection criterion to select the best HMM model size given data. From our discussion earlier, for a model λ trained on data X , the best

model size is the one, when coupled with its ML configuration, gives the highest posterior probability. From section 4.2, we know that $P(\lambda|X) \propto P(X|\lambda)$, and the marginal likelihood can be efficiently approximated using the BIC and the CS measures.

Applying the BIC approximation, marginal likelihood of the HMM, λ_k , for cluster k is computed as:

$$\log P(X_k|\lambda_k) \approx \sum_{j=1}^{N_k} \log P(X_{kj}|\lambda_k, \hat{\theta}_k) - \frac{d_k}{2} \log N_k,$$

where N_k is the number of objects in cluster k , d_k is the number of parameters³ in λ_k , and $\hat{\theta}_k$ is the ML parameters in λ_k .

Applying the CS approximation, the marginal likelihood of a HMM, λ_k , is computed as the sum of data likelihood and model prior probability. To compute model prior probability, we made certain assumptions about the prior probability distributions of the parameters. Given that the HMM for cluster k has m_k states, the set of initial state probabilities, π_i 's, and the transition probabilities from any single state i , a'_{ij} 's, can be regarded as following the Bernoulli distribution, i.e., $\pi_i \geq 0$, $\sum_{i=1}^{m_k} \pi_i = 1$, and $a_{ij} \geq 0$, $\sum_{j=1}^{m_k} a_{ij} = 1$, we assume that they follow the Dirichlet prior distribution: $P(a_{i1}, \dots, a_{im_k}|\lambda_k) = D(a_{i1}, \dots, a_{im_k}|\lambda_k)$ and $P(\pi_1, \dots, \pi_{m_k}|\lambda_k) = D(\pi_1, \dots, \pi_{m_k}|\lambda_k)$.

For parameters defining the emission distributions, we assume that the feature mean values in each state, μ_i , are uniformly distributed, and the standard deviations of each state, σ_i , follow Jeffery's prior distribution (Cheeseman & Stutz, 1996), i.e.,

$$\begin{aligned} P(\mu_f|\lambda_k) &= \frac{1}{\mu_{f_{max}} - \mu_{f_{min}}}, \\ P(\sigma_f|\lambda_k) &= \sigma_f^{-1} [\log \frac{\sigma_{f_{max}}}{\sigma_{f_{min}}}]^{-1}, \end{aligned}$$

where μ_f and σ_f represents the mean and standard deviation value for the f th temporal feature, and $\mu_{f_{max}}/\mu_{f_{min}}$ and $\sigma_{f_{max}}/\sigma_{f_{min}}$ are the maximum/minimum mean and standard deviation values for feature f across all clusters. Therefore,

$$\begin{aligned} \log P(\hat{\theta}_k|\lambda_k) &= \log P(\pi_1, \dots, \pi_{m_k}|\lambda_k) \\ &+ \sum_{i=1}^{m_k} [\log P(\alpha_1, \dots, \alpha_{im_k}|\lambda_k) \\ &+ \sum_{f=1}^F \log(P(\mu_f|\lambda_k) \cdot P(\sigma_f|\lambda_k))]. \end{aligned}$$

Figure 1 illustrates how BIC and CS work for HMM model size selection. Data generated on a 5-state HMM is modeled using HMMs of sizes ranging from 2

³Significant parameters include all the parameters for emission probability definitions and only the initial probabilities and transition probabilities that are greater than a threshold value t , t is set to 10^{-6} for all experiments reported in this paper.

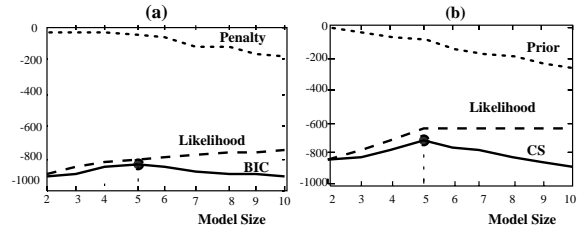


Figure 1. HMM model size selection

to 10. Results for the BIC and CS measures are plotted. The dashed lines show the likelihood of data for the different size HMMs. The dotted lines show the penalty (Figure 1 (a)) and the parameter prior probability (Figure 1 (b)) for each model. And the solid lines show BIC (Figure 1 (a)) and CS (Figure 1 (b)) as a combination of the above two terms. We observe that as the size of the model increases, the model likelihood also increases and the model penalty and parameter prior decreases monotonically. Both BIC and CS have their highest value corresponding to the size of the original HMM for data.

5.1.2 CRITERION FOR PARTITION SELECTION

In the Bayesian framework, the best clustering mixture model, M , has the highest *partition posterior probability* (PPP), $P(M|X)$. We approximate PPP with the marginal likelihood of the mixture model, $P(X|M)$. We compare the BIC and CS approximations of the marginal likelihood computation for cluster partition selection.

For partition with K clusters, modeled as $\lambda_1, \dots, \lambda_K$, the PPP computed using the BIC approximation is:

$$\begin{aligned} \log P(X|M) &\approx \sum_{i=1}^N \log [\sum_{k=1}^K P_k \cdot P(X_i|\hat{\theta}_k, \lambda_k)] \\ &- \frac{K + \sum_{k=1}^K d_k}{2} \log N, \end{aligned}$$

where $\hat{\theta}_k$ and d_k are the ML model parameter configuration and the number of significant model parameters of cluster k , respectively. P_k is the likelihood of data given the model for cluster k . When computing the data likelihood, we assume that the data is complete, i.e., each object is assigned to one known cluster in the partition. Therefore, $P_k = 1$ if object X_i is in cluster k , and $P_k = 0$ otherwise. The best model is the one that balances the overall data likelihood and the complexity of the entire cluster partition.

In the CS approximation, the independent variable C can be regarded as following a Bernoulli distribution, i.e., $P_k \geq 0$, $\sum_{k=1}^K P_k = 1$, again making the assumption that the prior distribution of C follows the Dirichlet distribution, we get

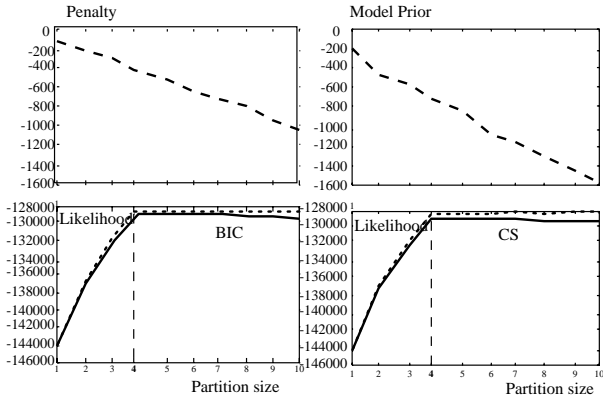


Figure 2. Cluster partition selection

$$\log P(X|M) \approx \log(P(P_1, \dots, P_K | \lambda_k) \cdot \prod_{k=1}^K P(\hat{\theta}_k | \lambda_k)) + \sum_{i=1}^N \log(\sum_{k=1}^K P_k \cdot P(X_i | \hat{\theta}_k, \lambda_k)),$$

where $\log P(\hat{\theta}_k | \lambda_k)$ s, are the prior probability of the ML parameter configuration for individual cluster models. Figure 2 illustrates how the BIC and the CS criteria work for cluster partition selection: given data consisting of an equal number of objects from four randomly generated HMMs, the BIC and the CS scores are measured when data is partitioned into 1 to 10 clusters. At first when the number of clusters is small, because the improvements of data likelihood dominates the change of the BIC and the CS values, values of both criteria monotonically increase as the size of the partition increases. Both criteria reach the peak value when the size of the partition corresponds to the true partition size, four. Subsequently, the improvements of data likelihood becomes less and less significant, the penalty on model complexity and the model prior terms dominate the change of the BIC and the CS measures, and both decrease monotonically as the size of the partition continues to increase.

5.2 The Clustering Search Control Structure

The exponential complexity associated with the four nested search steps for our HMM clustering prompts us to introduce heuristics into the search process. Figures 1 & 2 illustrated the characteristics of the BIC and the CS criterion in partition selection (step 1) and HMM model size selection (step 3). They both monotonically increase as the size of the HMM model or the cluster partition increase, until they reach a peak, and then begin to decrease because the penalty term dominates. The peak corresponds to the optimal model size. Given this characteristic, we employ the same sequential search strategy for both search steps 1 and 3. We start with the simplest model, i.e., a one clus-

ter partition for step 1 and a one state HMM for step 3. Then, we gradually increase the size of the model, i.e., adding one cluster to the partition or adding one state to the HMM, and re-estimate the model. After each expansion, we evaluate the model using the BIC or the CS measure. If the score of the current model decreases from that of the previous model, we may conclude that we have just passed the peak point, and accept the previous model as our final model. Otherwise, we continue with the model expansion process.

To expand a partition, we first select a seed based on the cluster models in the current partition. A seed includes a set of k objects, ($k = 3$ for all experiments shown here). The purpose of including more than one object in each seed is to ensure that there is sufficient data to build a reliable initial HMM. The first object in a seed is selected by choosing an object that has the least likelihood given all cluster models in the current partition. Then the remaining objects in the seed are the ones have the highest likelihood given the HMM model built based on the first object. A seed selected in this way is more likely to correspond to the centroid of a new cluster in the given partition. We apply HMM model size selection for each chosen seed.

Once a partition is expanded with a seed, search step 2 distributes objects to individual clusters such that the overall data likelihood given the partition is maximized. We assign object, x_i , to cluster, $(\hat{\theta}_k, \lambda_k)$, based on its sequence-to-HMM likelihood measure (Rabiner, 1989), $P(x_i | \hat{\theta}_k, \lambda_k)$. Individual objects are assigned to clusters whose model provides the highest data likelihood. If after one round of object distribution, any object changes its cluster membership, models for all clusters are updated to reflect the current data in the clusters. Then, all objects are redistributed based on the set of new models. Otherwise, the distribution is accepted.

After the objects are distributed into clusters, for a HMM model size, step 4 estimates the model parameters for each cluster using the Baum-Welch procedure discussed in section 2.2.

Table 1 gives the complete description of the sequential Bayesian HMM clustering (BHMMC) algorithm. In this algorithm, the HMM model size selection is not applied during object redistribution. This is because: when a single HMM is built based on data generated from k different HMMs, if there is no limit on the size of the model, the best model for the data may be a k -composite model, i.e., one that has the set of states equivalent to the combination of states from all k models. During BHMMC clustering, the intermediate clusters tend to include data from multiple models.

Table 1. The sequential BHMMC control structure

<pre> Initial partition construction: Select first seed Apply HMM model construction based on the seed Form the initial partition with the HMM do Partition expansion: Select one additional seed Apply HMM model construction based on the seed Add HMM to the current partition Object redistribution: do Distribute objects to clusters with the highest likelihood Apply HMM configuration for all clusters while there are objects change cluster memberships Compute PPP of the current partition while PPP of the current partition > PPP of the previous partition Accept the previous partition as the final cluster partition Apply HMM model construction to final clusters </pre>
--

To apply the HMM model construction for these intermediate clusters may create composite models that can give high data likelihood compared to HMM models built based on data from a seed. These composite models can be hard to split up in subsequent partitioning iterations. Because our goal is to learn the number of patterns exhibited in data and to study models characterizing individual patterns, in general, if k models are significantly different from each other, we prefer to build k separate, smaller, models, than to build one complex model that tries to include all k models. This is especially true when some of the models involved share a number of states, i.e., states appear in different models have very similar state definitions. In this case, given the complex, composite model, it is difficult to determine the set of submodels because the transition probabilities in and out of these shared states represent composite statistics accumulated from all models involved.

6. Experimental Results

First, we describe how synthetic models and data are generated for the experiments. Then, we give the performance indices we use to evaluate the experimental results. Finally, we analyze the experimental results using the proposed performance indices.

6.1 Data

To construct HMM models of different sizes, first, we assign state definitions by randomly selecting mean and variance values from value ranges $[0, 100]$ and $[0, 25]$ respectively. Then we assign state transition probabilities and initial probabilities by randomly sampling from value range $[0, 1]$, and then normalize the probabilities.

Based on each model, we randomly generated 30 objects, each object is described by two temporal features, and the sequence length of each feature is set to 100. For experiment 1, we generated five different HMMs for each of the three model sizes: 5, 10, and 15 states. Then, a separate data set is created based on each of these 15 HMMs. For experiment 2, we constructed five sets of HMMs, each set consists of four HMMs of different sizes, i.e., 3, 6, 9, and 12 states. For each set of HMMs, one data set is created by combining data objects generated from the 4 different HMMs. For these combined data sets, we know the number of models involved, and the model size and parameter configuration of each.

6.2 Performance Indices

In addition to the partition posterior probability, we propose two other performance indices to evaluate the quality of the cluster partitions generated:

Partition Misclassification Count (PMC) computes the number of object misclassification in the C_i cluster assuming G_i is the true cluster. C_i and G_i correspond when a majority of the C_i objects have the G_i label, and no other G_k ($k \neq i$) group has a larger number of objects in C_i . If more than three groups are formed, the additional smaller C groups are labeled as fragmented groups. The misclassification count is computed as follows: if an object falls into a fragmented C group, where its type (G label) is a majority, it is assigned a misclassification value of 1. If the object is a minority in a non-fragmented group, it is assigned a misclassification value of 2. Otherwise the misclassification value for an object is 0. When the true partition model is available, the smaller the sum of the misclassification counts for all objects in a partition, the better quality the partition in comparison.

Between Partition Similarity (BPS) measures the similarity between two partitions in terms of the likelihood of temporal sequences generated by one partition given the other partition, and vice versa. Given two partitions P_i and P_j , each with N_i and N_j number of clusters and models: $\lambda_1, \dots, \lambda_k, \dots, \lambda_{N_i}$, and $\lambda_1, \dots, \lambda'_k, \dots, \lambda_{N_j}$ respectively, the distance between the two partitions can be computed by: $BPS(P_i, P_j) = \frac{\sum_{k=1}^{N_i} \text{Max}_{\lambda'_{k'} \in P_j} P(S_i^k | \lambda'_{k'}) + \sum_{k'=1}^{N_j} \text{Max}_{\lambda_k \in P_i} P(S_j^{k'} | \lambda_k)}{N_i + N_j}$,

where S_i^k and $S_j^{k'}$ are temporal sequences generated based on the k th HMM in partition i and the k' th HMM in partition j respectively. For each partition, one temporal sequence is generated from the HMM of each cluster. The likelihood of the temporal

Table 2. BIC and CS for HMM model size selection

Criterion measure	True HMM model size		
	5	10	15
BIC	5(0)	10(0)	13.2(2.2)
CS	5(0)	10(0)	13.2(2.2)

Table 3. BIC and CS for cluster partition selection

Criterion measure	Fixed HMM size clustering			Varying HMM size clustering
	3	8	15	
BIC	48(24)	26(27)	70(22)	0(0)
CS	26(27)	24(29)	84(29)	0(0)

sequence given the other partition is approximated by the largest likelihood of the temporal sequence given all HMMs in the other partition. The overall BPS is the normalized sum of the likelihood of all temporal sequences from the two partitions. When computing the distance between one partition and the true partition, i.e., partition with the set of HMMs that generate the data, the larger the BPS, the more similar the partition in comparison to the true partition, thus the better the partition.

6.3 Experiments

The first experiment compares the effectiveness of the BIC and the CS measures in selecting HMM model sizes based on data. Table 2 shows average model sizes and the standard deviations of the HMMs derived from data. In all cases, the BIC and the CS measures selected HMM models of exactly the same sizes. For 5-state and 10-state HMMs, both measures selected HMMs that have sizes identical to the generative HMMs. For 15-state generative HMMs, the sizes of the derived models differ among trials, and have an average size smaller than that of the true HMMs. This is attributed to the well known problem with the Baum-Welch ML parameter estimation procedure, i.e., it sometimes converges to locally maximum parameter configuration, which prematurely terminates the sequential HMM model size search process.

The second experiment compares the performance of the BIC and the CS measures to cluster partition selection. It also studies the effect of the HMM model size selection on cluster partition generation. Two different clustering methods are used: (1) the sequential BHMMC which employs the dynamic HMM model size selection, and (2) fixed HMM size clustering which does not perform HMM model size selection. Instead,

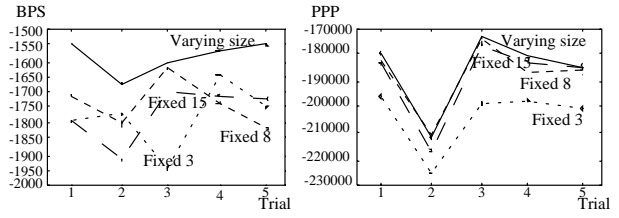


Figure 3. HMM cluster using HMM model size selection vs. using fixed HMM model size

a pre-determined, fixed size HMM is used throughout clustering.

Table 3 shows the PMC for partitions generated for different data. The cluster partitions selected by the BIC and the CS measures are not always identical, but they generally agree with each other. When model size selection is applied, both measures select the partitions that are perfect match to the partition with the set of generative models. When model size selection is not applied, for both measures, the partitions generated with too small a fixed HMM, i.e., a 3-state HMM, are considered better than those generated with too big a fixed HMM, i.e., a 15-state HMM. Both measures generated partitions of better quality when the fixed HMM size equals to the average size of the four generative HMMs.

When the size of the HMMs are fixed and small, they do not possess the ability to discriminate among objects that are generated from multiple, more complex HMMs. Therefore, objects from different generative HMMs are grouped into the same cluster in the final partition. On the other hand, when using fixed size HMMs that are too big, adding one new cluster to the partition incurs a large model complexity penalty that sometimes can not be offset by the data likelihood gain. Also, the problem of forming composite models arises when the fixed HMM size used is too large. In both cases, objects from different generative HMMs are mixed into the same cluster in the partition generated. When the HMM model selection procedure is applied, individual clusters are modeled with HMMs of appropriate sizes to best fit data, and the complexity of all HMMs in the partition and the overall data likelihood are carefully balanced. These lead to better quality cluster partitions. In addition to counting the misclassifications of each partition, we also compare the partition in terms of its PPP and BPS scores in Figure 3. The solid lines represent the sequential BHMMC, and the three dashed lines represent clustering with fixed 3-state HMM, 8-state HMM, and 15-state HMM. For all trials, partitions generated with HMM model size selection have higher

posterior model probability and larger between partition distance than those obtained from clustering with the fixed size HMMs.

7. Summary

We have presented a Bayesian clustering methodology for temporal data using HMMs. The clustering process incorporates a HMM model size selection procedure which not only generates more accurate model structure for individual clusters, but also improves the quality of the partitions generated. This creates a tradeoff between improved quality of the cluster models and the cluster partitions and increased computational complexity of the algorithm⁴. From the experimental results, we believe that the improvements are significant enough to make the extra computation worth while. We cast both HMM model size selection and cluster partition selection problems into the Bayesian model selection framework and have experimentally shown that the BIC and the CS measures are comparable when applied in both tasks.

The incorporation of heuristics into the search control structure have dramatically cut down the search space involved, but HMM clustering algorithm is still computationally complex. Therefore, we would like to employ incremental clustering strategies where we start with a cluster partition built based on small data, and gradually revise the size and structure of the partition as more and more data is collected. Also, we would like to look into other partition evaluation criterion that is based on model prediction accuracy. Even though the purpose of our HMM clustering is not prediction per se, how well the set of models can predict may be used to evaluate the quality of the partition.

References

- Cheeseman, P., & Stutz, J. (1996). Bayesian classification (autoclass): Theory and results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.), *Advances in knowledge discovery and data mining*, 153–180. Cambridge, MA: MIT press.
- Chib, S. (1995). Marginal likelihood from the gibbs sampling. *Journal of the American Statistical Association*, 90, 1313–1321.
- Dempster, A. P., Laird, N. M., & 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., & Kokkinakis, G. (1996). Algorithm for clustering continuous density hmm by recognition error. *IEEE Transactions on Speech and Audio Processing*, 4, 231–234.
- Heckerman, D., Geiger, D., & Chickering, D. M. (1995). A tutorial on learning with bayesian networks. *Machine Learning*, 20, 197–243.
- Kass, R. E., & Raftery, A. E. (1995). Bayes factor. *Journal of the American Statistical Association*, 90, 773–795.
- Kosaka, T., Masunaga, S., & Kuraoka, M. (1995). Speaker-independent phone modeling based on speaker-dependent hmm's composition and clustering. *Proceedings of the Twentieth International Conference on Acoustics, Speech, and Signal Processing* (pp. 441–444).
- Ostendorf, M., & Singer, H. (1997). Hmm topology design using maximum likelihood successive state splitting. *Computer Speech and Language*, 11, 17–41.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77, 257–285.
- Rabiner, L. R., Lee, C. H., Juang, B. H., & Wilpon, J. G. (1989). Hmm clustering for connected word recognition. *Proceedings of the Fourteenth International Conference on Acoustics, Speech, and Signal Processing* (pp. 405–408).
- Sebastiani, P., Ramoni, M., Cohen, P., Warwick, J., & Davis, J. (1999). Discovering dynamics using bayesian clustering. In D. J. Hand, J. N. Kok and M. R. Berthold (Eds.), *Advances in intelligent data analysis*, 199–210. Berlin, Springer-Verlag.
- Smyth, P. (1997). Clustering sequences with hidden markov models. In M. C. Mozer, M. I. Jordan and T. Petsche (Eds.), *Advances in neural information processing*, 648–654. Cambridge, MA, MIT Press.
- Stolcke, A., & 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. Berkeley, CA.
- Thiesson, B., Meek, C., Chickering, D. M., & Heckerman, D. (1998). *Learning mixtures of bayesian networks* (Technical Report MSR-TR-97-30). Microsoft Research, One Microsoft Way, Redmond, WA.

⁴For data in experiment two, BHMMC with HMM model size selection takes an average of 30 hours on a Sparc Ultra machine.