Multivariate Clustering by Dynamics

Marco Ramoni
Knowledge Media Institute
The Open University
Milton Keynes MK7 6AA, UK
m.ramoni@open.ac.uk

Paola Sebastiani
Department of Mathematics
Imperial College
London SW7 2B, UK
p.sebastiani@ic.ac.uk

Paul Cohen
Department of Computer Science
University of Massachusetts
Amherst MA 01003-4610
cohen@cs.umass.edu

Abstract

We present a Bayesian clustering algorithm for multivariate time series. A clustering is regarded as a probabilistic model in which the unknown auto-correlation structure of a time series is approximated by a first order Markov Chain and the overall joint distribution of the variables is simplified by conditional independence assumptions. The algorithm searches for the most probable set of clusters given the data using a entropy-based heuristic search method. The algorithm is evaluated on a set of multivariate time series of propositions produced by the perceptual system of a mobile robot.

Introduction

Suppose one has a set of time series generated by one or more unknown processes, and the processes have characteristic dynamics. Clustering by dynamics is the problem of grouping time series into clusters so that the elements of each cluster have similar dynamics. Suppose a batch contains a time series of stride length for every episode in which a person moves on foot from one place to another. Clustering by dynamics might find clusters corresponding to "ambling," "striding," "running," and "pushing a shopping cart," because the dynamics of stride length are different in these processes. Similarly, cardiac pathologies can be characterized by the patterns of sistolic and diastolic phases; economic states such as recession can be characterized by the dynamics of economic indicators; syntactic categories can be categorized by the dynamics of word transitions; sensory inputs of a mobile robot can be merged to form prototypical representations of the robot's experiences.

The task of clustering time series can be regarded as the process of finding the partition, i.e. the set of clusters, best fitting the data according to some criteria. Typically, this task involves two steps: (1) model each time series to capture its essential dynamical features; (2) partition the set of time series by clustering. Our approach uses one of the simplest representations of a time series: a first order Markov chain (MC). A MC assumes that the probability distribution of a variable at time t is independent of the variable values observed prior to time $t-1$ (Ross 1996). Furthermore, we regard the task of finding the best partition of the data as a statistical model selection process. Smyth (1999) applied this idea to clustering time series and Sebastiani *et al.* (1999) devised a Bayesian model-based algorithm to cluster higher-order MCs. The algorithm, called Bayesian Clustering by Dynamics (BCD), has been successfully applied to cognitive robotics (Sebastiani, Ramoni, & Cohen 2000; Cohen *et al.* 2000), simulated war games (Sebastiani *et al.* 1999), behavior of stock exchange indices, and unsupervised generation of musical compositions. These applications suggest that even a very simple MC representation of a dynamic process is powerful enough to capture common aspects of different time series. Furthermore, an appealing feature of BCD is an entropy-based heuristic that makes the search over the space of partitions very efficient. In its current formulation, BCD is limited to the univariate case, that is, the algorithm is able to cluster the behaviors of only one variable at a time. But what if the problem at hand is multivariate, that is, it is represented by simultaneous time series of several interacting variables? The assessment of a battlefield situation is done on the basis of several, possibly interacting, factors, like force ratio, number of engaged units, total forces mass, and so on. Similarly, a sensory experience of a mobile robot is given by the simultaneous values of several sensors, and the experience itself can be identified by the correlation among subsets of these variables.

Suppose one wants to cluster a set of multivariate time series of v discrete variables, each taking c values. The straightforward solution is to convert the problem into a univariate one by defining a single variable taking as values all combinations of values of the v variables and then applying the univariate case algorithm (Sebastiani *et al.* 1999). Unfortunately, this solution is hardly scalable because the number of states of this variable grows exponentially with the number of the original v variables. The solution we present in this paper is a novel clustering technique for multivariate time series, called Multivariate Bayesian Clustering by Dynamics (MBCD). The clustering algorithm is model-based, as it represents a clustering as a probabilistic model, it is Bayesian, as both the decision of whether grouping MCs and the stopping criterion are based on the clustering posterior probability, and it uses an entropy-based heuristic to reduce the search space over a subset of possible partitions. The clusters of dynamics produced by the algorithm are sets of MCs, which are assumed to be conditional independent given cluster membership. Thus, they capture dynamics involving simultaneously all the variables but the conditional independence assumption makes the algorithm scalable to large data sets. The algorithm is tested on multivariate time series of propositions produced by a mobile robot perceptual system and produces clusters which are significantly different from random clustering and in agreement with human clustering.

Theory

Suppose we have a set $S = \{S_1, \cdots, S_m\}$ of m multivariate time series. Each multivariate time series S_k is a set of v univariate time series $S_{k1}, \cdots, S_{kv}$ recording values of variables $X_1, ..., X_v$. The multivariate clustering algorithm can be outlined as follows. Given the $m \times v$ univariate time series, construct a MC for each series and replace each of the m multivariate time series by a set of v MCs. Rank the m sets of MCs in decreasing order of distance, merge similar sets of MCs into clusters if the merging increases a scoring metric, and repeat the procedure until a stopping criterion is met. The first step is the estimation of a MC from a univariate time series and it is considered next.

Markov Chains

Suppose that, for a variable X, we observe the time series $(x_0, x_1, x_2, ..., x_{i-1}, x_i, ..)$, where each x_i is one of the states $1, ..., s$ of X. The process generating the series is a (first order) MC if the conditional probability that the variable X visits state j at time t, given the sequence $(x_0, x_1, x_2, ..., x_{t-1})$, is only a function of the state visited at time $t-1$. Hence, we write $p(X_t = j|(x_0, x_1, x_2, ..., x_{t-1})) = p(X_t = j|x_{t-1})$, where X_t denotes the variable X at time t, and a MCs is represented by a table $P = (p_{ij})$ of transition probabilities, where $p_{ij} = p(X_t = j|X_{t-1} = i)$ is the probability of visiting state j given the current state i.

Given a time series generated from a MC, we might estimate the probabilities of state transitions $(i \rightarrow j) \equiv X_t = j|X_{t-1} = i$ from the data as $p_{ij} = n_{ij}/n_i$, where $n_i = \sum_j n_{ij}$ and n_{ij} is the frequency of the transitions $(i \rightarrow j)$ observed in the time series. Instead we prefer a Bayesian estimate in which prior information about transition probabilities can be taken into account. The derivation of this estimate is given in Sebastiani *et al.* (2000), here we simply give the result: The probability $\hat{p}_{ij}$ is estimated as

$$\hat{p}_{ij} = \frac{\alpha_{ij} + n_{ij}}{\sum_i \alpha_{ij} + n_{ij}} \qquad (1)$$

where the so called prior hyper-parameter α_{ij} can be thought of as the prior frequency of transition $(i \rightarrow j)$, thus encoding prior knowledge about the process. In particular, the ratio α_{ij}/α_i is the prior probability of transition $(i \rightarrow j)$ and $\hat{p}_{ij}$ is the posterior probability in the sense of being estimated from prior information α_{ij} and the observed frequency n_{ij} of transition $(i \rightarrow j)$.

Clustering

The first step of the algorithm replaces the original time series by MCs represented by transition probability tables, as shown in Figure 1. This conversion process transforms each multivariate time series S_k in S into a set P_k of transition probability matrices P_{kh}, one for each variable X_h. We can now cluster the m sets of transition probability matrices.

$$\begin{array}{cccc} S & X_1 & \cdots & X_v \\ \hline S_1 & S_{11} & \cdots & S_{1v} \\ \vdots & & \vdots & \\ S_m & S_{m1} & \cdots & S_{mv} \end{array}$$

$$\Downarrow$$

$$\begin{array}{cccc} S & X_1 & \cdots & X_v \\ \hline P_1 & P_{11} & \cdots & P_{1v} \\ \vdots & & \vdots & \\ P_m & P_{m1} & \cdots & P_{mv} \end{array}$$

Figure 1: The first step of the MBCD algorithm replaces time series S_{kh} with transition probability matrices P_{kh}.

As in BCD, the MBCD algorithm is *agglomerative*: it starts by assigning each set of transition matrices P_k to a separate cluster and iteratively merges them until a certain stopping criterion is met. Merging two sets of matrices $P_k = (P_{k1}, \cdots, P_{kv})$ and $P_l = (P_{l1}, \cdots, P_{lv})$ consists of creating a new set C_n of transition probability matrices $(P_{n1}, \cdots, P_{nv})$. The new cluster will be still a set of v transition matrices and each transition probability matrix P_{nh} in C_n is estimated from the cumulative transition frequencies of the variable X_h. The MBCD algorithm does not use a measure of similarity between MCs to decide whether two sets of MCs belong to the same cluster, neither it relies on a separate stopping rule. Both the decision of merging sets and the stopping criterion are based on the posterior probability of the obtained clustering, that is, the probability of the clustering given the data observed: Two sets of MCs are merged if the resulting partition has higher posterior probability than the partition in which these two sets are not merged, and the algorithm stops when no available merging produces a partition with higher posterior probability. MBCD's task is to find a maximum posterior probability partition of sets of MCs. Said in yet another way, MBCD solves a Bayesian model selection problem, where the model M_c it seeks is the most probable partition given the data. Details are given in the next section.

Method

Model selection methods are typically identified by two components: a scoring metrics — in this case, the posterior probability of a partition — and a search strategy to explore the space of possible partitions.

Posterior Probability

The key to the algorithm is the posterior probability of a partition. We regard a partition of the m sets of MCs into c clusters as a statistical model M_c, in which each cluster merges m_k sets of MCs. Components of this statistical model are the sets of MCs, a variable C with states $C_1, ..., C_c$ denoting cluster membership and a structure of dependency among the MCs in each cluster C_k. The variable C is a hidden, discrete variable, as it is not observed in the set S. The number c of states of C is unknown, but the number m of initial sets imposes an upper bound, as the number of clusters will be never higher than the number of initial multivariate time series. For example, if S is given by only two sets of multivariate time series S_1 and S_2, there are only two models describing possible partitions of these data: M_1 in which the two sets are merged into one cluster and M_2 in which the two sets are not merged. In model M_1, variable C takes one value while, in model M_2, C takes two values.

Globally, there are 2^m models describing different partitions. We can compute the posterior probability of these models by Bayes' Theorem:

$$p(M_c|S) = \frac{p(M_c)p(S|M_c)}{p(S)}$$

where $p(M_c)$ is a partition prior probability and $p(S)$ is the marginal probability of the data, and we choose the model with maximum posterior probability. Since we are comparing all models over the same data, $p(S)$ is constant and, for the purpose of maximizing $p(M_c|S)$, it is sufficient to consider $p(M_c)p(S|M_c)$. Furthermore, if all models are *a priori* equally likely, the comparison can be based solely on the *marginal likelihood* $p(S|M_c)$, which is a measure of how likely the data are if the model M_c is true. Reasonable assumptions on the sample space, the adoption of a particular parameterization for the model M_c and the specification of a conjugate prior lead to a simple, closed-form expression for the marginal likelihood $p(S|M_c)$ of which the solution presented in Sebastiani *et al.* (1999) is a special case for the univariate problem.

Conditional on the model M_c and hence on a specification of c clusters of sets of MCs, we suppose the marginal distribution of the variable C is multinomial, with cell probabilities $p_k = p(C = C_k)$. Furthermore, we suppose that sets of time series assigned to different clusters are mutually independent and that, for each cluster C_k, the v MCs generating the time series assigned to cluster C_k are independent. This last assumption says that, given cluster membership, the MCs are independent: Once we know the cluster we are in, the MC describing the dynamics of each variable X_h is independent of the dynamics describing any other variables. Thus, each cluster captures the overall dynamical features of a set of MCs that can however be treated as independent quantities within clusters. This assumption produces a simple expression for the probability of the data, given a probabilistic specification of model M_c. If we denote by $P_{kh} = (p_{khij})$ the transition probability matrix of the MC for variable X_h in cluster C_k, the probability of observing data S, given the set of probabilities $\theta = (p_k, p_{khij})$ is

$$p(S|\theta) = \prod_{k=1}^{c} p_k^{m_k} \prod_{h=1}^{v} \prod_{ij=1}^{s} p_{khij}^{n_{khij}}$$

where n_{khij} denotes the frequency of transition $(i \to j)$ observed in all the time series generated by the MC with transition probability matrix P_{kh} in cluster C_k, and m_k is the number of sets of time series assigned to cluster C_k. The probability $p(S|\theta)$ is derived as follows. The quantity $\prod_{ij=1}^{s} p_{khij}^{n_{khij}}$ is the probability of observing the transition $(i \to j)$ with frequency n_{khij} in a time series generated from a MC with transition probability matrix P_{kh}. In other words, $\prod_{ij=1}^{s} p_{khij}^{n_{khij}}$ is the probability of observing the time series assigned to the hth component of cluster C_k. Since MCs in a cluster C_k are independent, the probability of the data observed in the whole cluster C_k is computed as the product $\prod_{h=1}^{v} \prod_{ij=1}^{s} p_{khij}^{n_{khij}}$. The joint probability of the data is then computed by simply multiplying these quantity over all clusters defining the partition M_c.

Since quantity $p(S|\theta)$ is a function of the unknown parameter vector θ, in order to maximize the marginal likelihood $p(S|M_c)$, we need to average these unknown parameters out by using prior information. A standard Bayesian solution is to adopt sets of independent Dirichlet prior distributions with hyper-parameters α_{khij}, one Dirichlet for each conditional distribution $(p_{khij})_j$, and one Dirichlet distribution with hyper-parameters β_k for the distribution (p_k) over the clusters. The hyper-parameters α_{khij} and β_k encode prior knowledge about the probabilities p_{khij} and p_k in terms of the ratios $\alpha_{khij}/\sum_j \alpha_{khij}$ and $\beta_k/\sum_k \beta_k$. With this prior specification, the marginal likelihood $p(S|M_c)$ is given by

$$p(S|M_c) = \frac{\Gamma(\sum_k \beta_k)}{\Gamma(\sum_k \beta_k + m)} \prod_{k=1}^{c} \frac{\Gamma(\beta_k + m_k)}{\Gamma(\beta_k)} \times$$

$$\prod_{k=1}^{c} \prod_{h=1}^{v} \prod_{i=1}^{s_h} \frac{\Gamma(\sum_j \alpha_{khij})}{\Gamma(\sum_j [\alpha_{khij} + n_{khij}])} \prod_{j=1}^{s_h} \frac{\Gamma(\alpha_{khij} + n_{khij})}{\Gamma(\alpha_{khij})}$$

where $\Gamma(\cdot)$ is the Gamma function. When $v = 1$, that is, there is only one variable and S is a set of univariate time series, $p(S|M_c)$ equals the expression of the marginal likelihood provided by Sebastiani *et al.* (2000). Once the *a posteriori* most likely partition has been selected, the set of transition probability matrices P_{kh} associated with the cluster C_k can be estimated as

$$\hat{p}_{khij} = \frac{\alpha_{khij} + n_{khij}}{\sum_j [\alpha_{khij} + n_{khij}]}$$

and the probability of $C = C_k$ can be estimated as

$$\hat{p}_k = \frac{\beta_k + m_k}{\sum_k \beta_k + m}.$$

The marginal likelihood is a function of the hyper-parameters α_{khij} and β_k. A convenient choice is to set the initial $a = m \times \prod_h s_h^2$ hyper-parameters α_{khij} equal to α/a. In this way, the specification of the prior hyper-parameters requires only the prior precision α, which measures the overall confidence in the prior model. An analogous procedure can be applied to the hyper-parameters β_k associated with the prior estimates of p_k.

Heuristic Search

The number of possible partitions grows exponentially with the number of multivariate time series, and a brute force

Similarity Measure

$$
\begin{array}{cccccccc}
S & X_1 & & \cdots & & X_v & & D \\
\hline
S_k & P_{k1} & & \cdots & & P_{kv} & & \\
S_l & P_{l1} & & \cdots & & P_{lv} & & \\
 & \downarrow & & & & \downarrow & & \\
 & D_{kl1} & + & \cdots & + & D_{klv} & \rightarrow & D_{kl}
\end{array}
$$

Figure 2: Computation of the similarity measure between two sets of MCs.

search in the set of partitions would be infeasible. We introduce here a similarity-based heuristic search. The intuition behind the heuristic is that we have better chances of increasing the marginal likelihood when we merge *similar* clusters. Therefore, if we merge first more similar clusters, we have better chances of reaching the maximum posterior probability partition sooner.

To implement this method, we need to define a measure of similarity able to guide the search process. Our approach is sketched in Figure 2. Recall that each set P_k is a collection of v transition probability matrices P_{kh}, one for each variable X_h. Transition probability matrices P_{kh} and P_{lh} in two sets P_k and P_l are comparable only when they refer to the same variable X_h, and rows with the same index are probability distributions conditional on the same event and they are, therefore, comparable. Thus, a measure of similarity between two sets of MCs can be constructed by evaluating a row-by-row distance between pairs of comparable transition probability tables P_{kh} and P_{lh} and then by summarizing this row-by-row distance for all tables. The measure of similarity currently used by MBCD is an average of the Kullback-Liebler distances between comparable tables so that, by letting p_{khij} and p_{lhij} be the probabilities of transition $(i \rightarrow j)$ in P_{kh} and P_{lh}, the Kullback-Liebler distance of the two probability distributions in row i is $D_{klhi} = \sum_j p_{khij} \log(p_{khij}/p_{lhij})$. The average distance between P_{kh} and P_{lh} is then $D_{klh} = \sum_i D_{klhi}/s_h$, with s_h denoting the number of states of variable X_h, and the overall distance between the two sets S_k and S_n is $\sum_h D_{klh}$.

Iteratively, MBCD computes all pairwise distances between sets of transition probability tables, sorts the generated distances, merges the two closest sets and evaluates the result. The evaluation asks whether the new model M_c, in which two sets of MCs are merged, is more probable than the model M_{c+1} in which these sets are separated, given data S. If the probability $p(M_c|S)$ is larger than $p(M_{c+1}|S)$, MBCD replaces the two sets of MCs with the cluster resulting from their merging. Then, MBCD updates the set of ordered distances and repeats the procedure on the reduced set space. If the probability $p(M_c|S)$ is not larger than $p(M_{c+1}|S)$, MBCD tries to merge the second best, the third best, and so on, until no further merging is possible and, in this case, MBCD returns the most probable partition found so far. Note that the similarity measure is just used as a heuristic guide for the search process rather than a grouping criterion.

Evaluation

We evaluated the technique four ways. First, we compared the partitions found by MBCD with those produced by another clustering technique and by a human judge. Second, we developed a measure of the quality of partitions and showed that MBCD partitions score significantly higher than random partitions. Third, we examined MBCD partitions by hand to see whether they make sense. Fourth, we tested MBCD with different values of the prior precision parameter. In previous work, Schmill *et al.* (1999) constructed a set of 102 trials in which a Pioneer 1 robot interacted with objects in its environment, moving toward or past objects, pushing them, reversing away from them, and so on. Each trial produced two qualitatively different kinds of multivariate time series: a series of a vector of continuous values from roughly 40 sensors, and a series of symbolic states. States are lists of symbolic propositions; for example, ((:STOP :R) (:IS-RED :A) (:IS-OBJECT :A)) is a state in which the robot is stopped and perceives a red object. Schmill *et al.* clustered the 102 trials in the data set by hand, and also ran a clustering algorithm based on dynamic time warping on the sensor time series, and compared the two sets of clusters S_1 and S_2 as follows: For every pair of trials i and j, record an *agreement* if i and j reside in a single cluster in S_1 and also reside in a single cluster in S_2, or i and j reside in different clusters in S_1 and also reside in a different cluster in S_2. The *concordance* of S_1 and S_2 is just the total number of agreements divided by the total number of pairs of trials. Schmill *et al.* report concordances greater than 0.9 in a variety of conditions. We apply the same procedure to the same set of 102 trials, computing concordances between MBCD and human clustering, and MBCD and clustering based on dynamic time warping. The results are very good. For the partition MBCD produces with prior precision equal to one, the concordance between MBCD's partition and the human partition is 0.82, and the concordance between MBCD's partition and the dynamic time warping partition is 0.81. To test whether these numbers are significant, we devised a randomization procedure to answer the question, "what is the expected value of the concordance statistic under the null hypothesis that the trials in a partition are distributed in random clusters?" Suppose a partition contains c clusters and each cluster i contains n_i trials. A random partition is generated by randomly selecting (without replacement) trials to construct c clusters of sizes $n_1 \ldots n_c$. The expected value of the concordance between a partition p and a random partition is easily obtained by generating a few hundred random partitions r_i, recording the concordance between p and r_i, and taking the mean of the resulting distribution of r_i. For the partition produced with prior precision equal to one, the expected value of the concordance with a random partition is .73, and the standard deviation of the distribution of r_i is .004. There is essentially no chance that a concordance between two partitions greater than .8 could arise if one of the partitions was random.

Any measure of the quality of a partition must reflect the principle that the similarity of items within clusters is high relative to the similarity of items in different clusters. MBCD clusters episodes, so we need a measure of the sim-

Cluster	Mean Similarity	Critical value
1	.5339	.4430
2	.4974	.4008
3	.5780	.4070
4	.7124	.4540
5	.5526	.3741
6	.4696	.3770
7	.6603	.3980

Table 1: Significance values with prior precision $\alpha = 1$.

ilarity of episodes, and ideally it should not be identical to the Kullback-Leibler metric that guides MBCD's search for partitions, because by design MBCD performs well according to that metric. Recall that each state in an episode is a set of propositions (e.g.,((:MOVING-BACKWARD :R) (:IS-RED :A) (:IS-OBJECT :A))). We say a proposition is *frequent* in an episode if the proposition appears in at least $p\%$ of the episode. Let P_1 and P_2 be sets of propositions that are frequent in episodes 1 and 2, respectively. Then a simple measure of similarity is the proportion of propositions frequent in either episode that are frequent in both episodes:

$$s = \frac{||P_1 \cap P_2||}{||P_1 \cup P_2||}.$$

Let $\bar{s}_i$ be the mean s for all pairs of episodes in cluster i, and let n_i be the number of episodes in that cluster. We want to know whether, for all clusters i, $\bar{s}_i$ is significantly higher that expected under the null hypothesis that MBCD performs no better than an algorithm that builds clusters by grouping randomly-selected episodes. The hypothesis is tested for each cluster by a simple randomization procedure: For cluster i, for c from 1 to k, select n_i episodes at random and calculate $\bar{s}_{i,c}$. The distribution of k values of $\bar{s}_{i,c}$ serves as a sampling distribution of $\bar{s}_i$ under the null hypothesis that clustering is random. The upper 99th percentile of this distribution, $\bar{s}_{i,0.99}$ serves as a critical value; if $\bar{s}_i > \bar{s}_{i,0.99}$, we reject the null hypothesis that cluster i was formed by a random algorithm, with $p \leq .01$. Values of $\bar{s}_i$ and $\bar{s}_{i,0.99}$ for a run of MBCD with prior precision $\alpha = 1$ are shown in Table 1. All of the results are highly significant, MBCD is performing much better than a random algorithm.

Qualitatively, the clusters produced by MBCD seem to make sense. Consider the seven clusters produced with prior precision equal to one. The first involves nine trials in which the robot moves toward objects A and C, sometimes bumping A, sometimes C. The second cluster, ten trials, involves the robot moving backwards. In four of these trials, C is the only object in view; in the other six trials, A becomes visible during the trial (by reversing, the robot increases the number of objects visible in its view). The third cluster is a bit of a mess: In all 17 trials, the robot moves forward and object C remains visible throughout. In ten trials, the robot bumps C; in seven trials, object B becomes visible during the trial; and in two trials, object A makes an appearance. In cluster 4, the robot moves forward, all three objects appear, with C in front of B and A appearing late (four trials); or A doesn't appear (two trials). In cluster 5, the robot approaches A in each of 24 trials, and bumps it in 11 trials. Object C enters the picture during 11 trials but disappears before the end of the trial. Cluster 6 incorporates 18 trials in which the robot is either moving forward or moving backward with no object in view, as well as three trials in which either A or C are in view. Cluster 7 also includes forward and backward movement (6 and 9 trials, respectively) in which object A remains in view, appears, or disappears (3, 3, and 9 trials, respectively). Finally, we ran MBCD with several values of the prior precision parameter. Qualitatively, increasing prior precision yields partitions with slightly larger numbers of clusters; the smallest partition contained seven clusters, the largest, sixteen. However, episodes that are grouped together under one value of prior precision are overwhelmingly likely to be grouped together under another: The average concordance of partitions across different values of prior precision is roughly 0.95.

Related and Future Work

Essential features of the MBCD algorithm are the model used to describe univariate time series, the intra-cluster structure of dependency, the model-based Bayesian approach to clustering and the heuristic search. These four features together make the algorithm different from other approaches to clustering multivariate time series.

MBCD uses first order MCs: the simplest model for a time series. More complex models involve the use of k-order MCs (Saul & Jordan 1999), in which the memory of the time series is extended to a window of k time steps, or Hidden Markov Models (Rabiner 1989), in which hidden variables are introduced to decompose the unknown auto-regressive structure of the time series into smaller components. MBCD can be easily extended to cluster sets of k-order MCs by modifying the expression for the marginal likelihood. Clustering multivariate time series modeled as Hidden Markov Models was considered by Smyth (1997) and future work will compare this approach with MBCD, to point out when one approach is preferable to the other. In another approach known as Dynamic Time Warping (Berndt & Clifford 1996), a series is stretched and compressed within intervals to make it fit the other as well as possible. The intuition behind the method is to cluster time series that have similar shapes. The work has recently been extended by Oates *et al.* (1999) to cluster time series generated from sensory inputs of a mobile robot. Our results, although limited to a single application, showed agreement between the two clustering methods and we are planning a more comprehensive comparison.

Closer to our approach is model-based clustering, originally developed by Banfield & Raftery (1993) to cluster static data and then applied by Smyth (1999) to time series. The probabilistic model used to represent multivariate clustering and its heuristic search sets MBCD apart. The rationale of the heuristic search is that merging sets of similar MCs first should result in better models and increase the posterior probability earlier in the search process. Although the current implementation of MBCD uses the Kullback-Liebler distance to build a similarity measure, other distances could be used and we are currently exploring the effect of replacing the Kullback-Liebler distance with other distance measures.

Conclusions

This paper presented the MBCD algorithm to cluster sets of multivariate time series. The algorithm regards a multivariate time series as a set of univariate time series, models each univariate time series as a MC and clusters sets of MCs using an entropy based heuristic search and a Bayesian scoring metric. An evaluation on a set of multivariate time series showed that the algorithm produces clusters which are significantly different from random clustering and in agreement with human clustering.

Acknowledgments

This research is supported by DARPA/AFOSR under contracts Nos F49620-97-1-0485 and N66001-96-C-8504. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA/AFOSR or the U.S. Government.

References

Banfield, J. D., and Raftery, A. E. 1993. Model-based gaussian and non-gaussian clustering. *Biometrics* 49:803–821.

Berndt, D. J., and Clifford, J. 1996. Finding patterns in time series: A Dynamic programming approach. In Fayyad, U. M.; Piatestsky-Shapiro, G.; Smyth, P.; and Uthurusamy, R., eds., *Advances in Knowledge Discovery and Data Mining*. Cambridge, MA: MIT Press. 229–248.

Cohen, P.; Ramoni, M.; Sebastiani, P.; and Warwick, J. 2000. Unsupervised clustering of robot activities: A Bayesian approach. In *Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000)*. New York, NY: ACM Press.

Oates, T.; Schmill, M. D.; and Cohen, P. R. 1999. Identifying qualitatively different experiences: Experiments with a mobile robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*. San Mateo, CA: Morgan Kaufman.

Rabiner, L. 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2):257–285.

Ross, S. M. 1996. *Stochastic Processes*. New York, NY: Wiley.

Saul, L. K., and Jordan, M. I. 1999. Mixed memory markov models: Decomposing complex stochastic processes as mixture of simpler ones. *Machine Learning* 37:75–87.

Schmill, M. D.; Oates, T.; and Cohen, P. R. 1999. Learned models for continuous planning. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics (Uncertainty 99)*. San Mateo, CA: Morgan Kaufman. 278–282.

Sebastiani, P.; Ramoni, M.; Cohen, P.; Warwick, J.; and Davis, J. 1999. Discovering dynamics using Bayesian clustering. In *Proceedings of the Third International Symposium on Intelligent Data Analysis (IDA-99)*. New York, NY: Springer. 199–209.

Sebastiani, P.; Ramoni, M.; and Cohen, P. 2000. Bayesian analysis of sensory inputs of a mobile robot. In *Case Studies in Bayesian Statistics*. New York, NY: Springer.

Smyth, P. 1997. Clustering sequences with hidden Markov models. In Mozer, M.; M.I.Jordan; and T.Petsche., eds., *Advances in Neural Information Precessing*. Cambridge, MA: MIT Press. 72–93.

Smyth, P. 1999. Probabilistic model-based clustering of multivariate and sequential data. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics (Uncertainty 99)*. San Mateo, CA: Morgan Kaufman. 299–304.