Clustering Time Series from Mixture Polynomial Models with Discretised Data

A. J. Bagnall, G. Janacek and M. Zhang

Technical Report CMP-C03-17
School of Computing Sciences
University of East Anglia
Norwich
England NR47TJ
Contact email: ajb@cmp.uea.ac.uk
Last Modified 24th September 2003

Abstract. Clustering time series is an active research area with applications in many fields. One common feature of time series is the likely presence of outliers. These uncharacteristic data can significantly effect the quality of clusters formed. This paper evaluates a method of overcoming the detrimental effects of outliers. We describe some of the alternative approaches to clustering time series, then specify a particular class of model for experimentation with k-means clustering and a correlation based distance metric. For data derived from this class of model we demonstrate that discretising the data into a binary series of above and below the median improves the clustering when the data has outliers. More specifically, we show that firstly discretisation does not significantly effect the accuracy of the clusters when there are no outliers and secondly it significantly increases the accuracy in the presence of outliers, even when the probability of outlier is very low.

1 Introduction

The clustering of time series has attracted the interest of researchers from a wide range of fields, particularly from statistics [36], signal processing [16] and data mining [11]. This has resulted in the development of a wide variety of techniques designed to detect common underlying structural similarities in time dependent data. A review of some of the work in the field is given in Section 2. These techniques have been applied to data arising from many areas, for example: web mining [12, 4]; finance and economics [42, 19]; medicine [22]; meterology [7]; speech recognition [16, 27]; gene expression analysis [17, 5] and robotics [44]. Our interest is primarily motivated by the desire to be able to detect common patterns of behaviour in bidding strategies of agents competing in markets in order to quantify adaptive agent performance [3].

Clustering is an unsupervised learning task, in that the learning algorithm is not informed whether the assignment of a data to a cluster is correct or not. For background into clustering see [24]. There are two main ways clustering has

been used with time series. Firstly, clustering can be applied to a single time series, frequently using a windowing system, to form different generating models of the single series [14]. Note that there is some controversy over the usefulness of this approach (see [26]). The second problem involves forming k clusters for m time series rather than from a single series. We are interested in the latter problem, which can be described as follows:

Given m time series, $S = \{s_1, \ldots, s_m\}$ of length $n_1, \ldots n_m$, the problem is to form k clusters or sets of time series, $C = \{C_1, \ldots, C_k\}$, so that the most "similar" time series are in the same cluster. k may or may not be known *a priori*. Also, cluster membership may be deterministic or probabilistic. To encompass both we can generalise the clustering task to assigning a probability distribution $p_s(j)$, to each time series which defines the probability that series s is in cluster C_j.

The obvious crucial question is what is meant by most "similar" time series. This is usually model and problem dependent, but can be generalised as follows. Suppose a distance function $d(a, b)$ is defined on the space of all possible series, D. A distance function $d(s_i, s_j) : D \times D \rightarrow \Re$ is a **metric** if it satisfies the four conditions

$$\begin{aligned} &d(a,b) > 0 && \text{if } a \neq b \\ &d(a,a) = 0 \\ &d(a,b) = d(b,a) \end{aligned}$$

$$d(a,c) \leq d(a,b) + d(b,c) \; \forall \;\; a, b, c \in A.$$

The distance function may have a domain that is the space of all time series or it may be embedded in a lower dimensional space formed through, for example, fitting a parameterised model to each series. Given a distance metric, the clustering task is to find the clusters that minimize the distance between the elements within each cluster and maximize the distance between clusters. This can be described by the introduction of a cost function. Suppose the cost for a cluster C_j is defined as

$$c_j = \sum_{a,b \in S} p_a(j) \cdot p_b(j) \cdot d(a,b).$$

The clustering problem for a given k is to find the partition that minimizes the total cost, $c = \sum_{j=1}^{m} c_j$. If k is not given, then some weighting function has to be included to encourage parsimonious clustering.

Clustering algorithms can be classified as two types, hierarchical methods and partitioning methods. Hierarchical methods involve calculating all distances then forming a dendrogram by using a linkage method such as nearest or furthest neighbour. The second approach involves partitioning using an iterative algorithm that attempts to optimize cluster assignation based on minimizing a cost function.

The most commonly using partitioning method is the k-means algorithm [34]. This is an iterative local search method that attempts to minimize the distance

```
Choose the k cluster centroids randomly
while the convergence criteria is not met
    Assign each data to the closest cluster centroid
    Recompute the cluster centroids using the current membership
```

within the clusters. Assuming the number of clusters required, k, is known, then k-means can be summarised as

The EM (Expectation Maximizing) algorithm is a generalisation of k-means [15]. Instead of assigning a data to a particular cluster, a probability of membership of all clusters is maintained. In addition to a centroid recording the means of the cluster, the EM algorithm also records a covariance matrix. A further alterna-

```
Initialise means and covariance matrix
while the convergence criteria is not met
    Compute the probability of each data belonging to each cluster
    Recompute the cluster distributions using the current
        membership probabilities
```

tive is the k-medoids method, which bases centroids on the median values rather than the means. A comprehensive description of clustering algorithms can be found in [24].

The aim of this research is to demonstrate that discretizing time series data can make clustering time series more robust to the presence of outliers without significantly decreasing accuracy when outliers are highly unlikely. Our initial approach to this is to define a simple class of underlying model similar to that used by other researchers then measure the effect on performance of the introduction of outliers. Section 2 provides background into some of the research into clustering time series. Section 3 describes experimentation on simulated data with a standard clustering technique (k-means) and a simple distance metric based on correlation and provides evidence of a scenario under which clipping allows the optimal clusters to be found. Section 4 summarises the results and describes the next stages of this research.

2 Related Research

Most research assumes some underlying form of the model and performs the clustering based on this assumption. [11] makes the case for a model based, or *gener-*

ative approach, which can be classified into three broad categories, discussed in Section 2.1: AutoRegressive Moving Average (ARIMA) models, Markov Chain and Hidden Markov models (MC and HMM) and polynomial mixture models. Approaches that do not assume a model form, often called similarity based approaches, are summarised in Section 2.2. Focardi [19] provides good background material on clustering time series.

2.1 Model Based Approaches

ARIMA Models: The main approach of statistics researchers to the problem of clustering time series is to assume the underlying models are generated by an ARIMA process [9]. The clustering procedure usually involves

1. fitting a model to each time series,
2. measuring distance between fitted models and
3. Clustering based on these distances.

This approach is adopted by Piccolo [43], Maharaj [35, 36] and Baragona [6]. Tong and Dabas [47] cluster different ARIMA models that have been fitted to the same data set, but the techniques used are also relevant to clustering models from different data sets.

Fitting the model requires the estimation of the structure and parameters of an ARIMA model. Structure is either assumed to be given or estimated using, for example, Akaike's Information Criterion or Schwartz's Bayesian Imformation Criterion [9]. Parameters are commonly fitted using the generalised least squares estimators. An order m ARIMA model can be fully specified by a set of parameters

$$\pi = \{\pi_1, \pi_2, \ldots \pi_m\}.$$

Some of the research based on assuming an ARIMA model derives the distance function from the differences between the estimates of these parameters. Piccolo [43] uses the Euclidean distance between the parameters,

$$d(\pi_a, \pi_b) = \left(\sum_{i=1}^{\infty} (\pi_{i,a} - \pi_{i,b})^2 \right)^{\frac{1}{2}}.$$

Maharaj [35, 36] adjusts her measure of distance between parameter sets by the correlation matrix estimated by the least squares to allow for dependent time series. She uses the resulting statistic as a test for
H_0: $\pi_a = \pi_b$ vs
H_1: $\pi_a \neq \pi_b$
A function of the p-value of this test is used as a similarity mesasure (low p-values making common cluster membership unlikely). An alternative approach to forming a distance function, used in [47, 6], is to base distance on the residuals of the model. Let e_a be the residuals for model π_a and $\rho_{a,b}(i)$ be the correlation between the residuals e_a and e_b. Tong [47] uses the sample correlation coefficient with lag 0, denoted $\rho(0)$,

$$d(a,b) = 1 - |\rho(a,b)(0)|$$

Baragona [6] uses a distance function that scales the zero lag correlation by the sum of the lagged correlations,

$$d(a,b) = \sqrt{\frac{(1-\rho^2_{a,b}(0))}{\sum_{i=1}^{m}(\rho^2_{a,b}(i))}}$$

This function was proposed in [8], although in this case it was used on the time series rather than the residuals of the models. A variety of clustering techniques have been employed. For example, principle coordinates and multidimensional scaling were used in [47, 43], hierarchical clustering with average linkage was used in [35] and with single linkage, complete linkage and Ward's method in [47] and heuristic search techniques (genetic algorithms, simulated annealing and tabu search) were employed in [6].

Hidden Markov Models (HMM): An alternative approach to the problem has been adopted by researchers in speech recognition and machine learning. Instead of an ARMA model, it is common to assume that the underlying generating models for each cluster can be accurately described as a markov chain (MC) or hidden markov model (HMM). A HMM is a set of unobserved states, each of which has an associated probability distribution for the random variable being observed, and a transition matrix that specifies the probability of moving from one state to another on any time step. A first-order HMM is an HMM where T is dependent only on the previous state. A MC also involves a set of states, except that the states correspond to the set of observable values of the random variable (and hence are not hidden).

For both approaches, the clustering algorithm generally involves the following steps:

1. form an initial estimate of cluster membership;
2. form HMM models based on membership;
3. while there is some improvement in models
   (a) adjust cluster membership;
   (b) reform models;

The clustering may be hierarchical or partitional. One key difference in technique between the ARIMA and the MC/HMM methods is that the ARIMA approach is to fit a model to each data before clustering, whereas most research into HMMs involves forming the cluster models on each iteration of the clustering algorithm.

MC models have been adopted by Ramoni et al [44] to model and cluster discrete series. Each state is associated with each value a data can take, and the problem becomes one of finding k transition matrices and identifying which series originates from which matrix. Their algorithm, called Bayesian Clustering by Dynamics (BCD), is a bottom up hierarchical agglomerative method, and involves the following steps:

1. Assume each time series is in its own cluster;
2. fit an MC to each series;
3. while new cluster models more likely than old cluster models;
   (a) merge the closets MC models;
   (b) reform models;

Distance between models is measured using the Kullback-Leiber distance.

Cadez *et al* [11] also use a MC model in the context of a generalised probabilistic EM based framework for clustering. In [12] they apply the technique to web mining. Ridgeway [45] compares using EM against Gibbs resampling when clustering Markov processes.

Smyth [46] clusters using HMM by fitting a model to each series, then uses the log-likelihood as a distance for a hierarchical furthest neighbour technique. Parameters for a given model structure are estimated with the Baum-Welch procedure.

Oates *et al* [39, 38, 41, 40] fit k HMMs using the Viterbi algorithm to train HMM on greedily selected subsets of series. In [40] they set the initial clustering using Dynamic Time Warping. HMM are fitted to each cluster, a Monte Carlo simulation is conducted on each model and series that are empirically unlikely to have been observed from a model are removed from the cluster. The model is then retrained and the process repeated until no more series can be removed. It is then tested whether unassigned series can be placed into other clusters. If not, they form their own, new clustering. They find that the hybridization of DTW and HMM forms better clusters than either approach alone on simulated data (which is also discretised) from models used in [46].

Zhong and Ghosh [52, 49–51, 48] use a model-based k-means clustering algorithm and a version of the EM algorithm. The also use a hierarchical model similar to that of [44], using HMM instead of MC models. Li and Biswas [31, 32, 29, 30, 33] propose a Bayesian HMM clustering methodology that includes determining the number of clusters and the structure of the HMM. Cadez, Gaffney and Smyth [11, 12] use HMM within the context of a generalised probabilistic framework. Alon *et al* [2] use the EM algorithm in HMM based clustering and assess the performance of EM in relation to k-means.

Polynomial Models Another approach is to assume the underlying model is a mixture of Polynomial functions. Gaffney and Smyth [20, 21] assume a mixture regression model. The EM algorithm with MAP estimates is used to estimate the cluster membership probabilities and weighted least squares used to fit the models. The technique is applied to simulated data, environmental data and video streaming data.

Bar-Joseph *et al* [5] adopt a mixture spline model for gene expression data, again using the EM algorithm in conjunction with least squares.

2.2 Model Free Approaches

Rather than assume a model form and base similarity on fitted parameter estimates, an alternative approach is to measure distance with the original or transformed data.

The simplest approach is to treat the time series as an N-dimensional vector and use the L_q Minkowsky distances. If a and b are series with N data and a_i, b_i are data at time i, then

$$L_q = (\sum_{i=1}^{N} |a_i - b_i|^q)^{\frac{1}{q}}$$

and

$$L_\infty = \max(|a_i - b_i|)$$

The Euclidean distance metric is L_2

$$L_2 = (\sum_{i=1}^{N} |a_i - b_i|^2)^{\frac{1}{2}} \tag{1}$$

This measure is used by [1] in conjunction with fast fourier transforms. The main problem with using an L_q measure for time series similarity is that they are effected by the scale of the two time series, thus shape characteristics can be lost (A further problem is that it is required that data be available for the same time steps, and this may not always be the case). [28] use a distance metric based on the Euclidean distance but introducing an extra set of shape parameters. An alternative is to use a metric that does capture the similarity in shape, for example one based on the correlation between the series. If we let $C(a, b)$ be the correlation between the series a and b, i.e.

$$C(a,b) = \frac{\sum_{i=1}^{N}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{N}(a_i - \bar{a})^2 \sum_{i=1}^{N}(b_i - \bar{b})^2}}$$

then Equation 2 is a metric, as demonstrated by Ormerod and Mounfield [42]. Similar metrics were used by [8].

$$d(a,b) = \sqrt{2(1 - C(a,b))} \tag{2}$$

Other researchers look for commonality measures based on common subsequences. For example [13] and [18] define measures based on common subsequences.

An alternative approach is to transform the data then use an associated metric. Approaches used include: time warping [40]; fast fourier transforms [1]; wavelet transforms [37] and piecewise constant approximation [25].

3 Experimentation

The results presented in this paper demonstrate that, for a certain class of underlying clustering model (described in Section 3.1), and with a particular experimental set up and clustering algorithm (outlined in Section 3.2), transforming the continuous time series into a discrete binary series

- does not significantly degrade clustering performance when there are no outliers and
- significantly improves the quality of the final clusters found when there are outliers, even when the probability of an outlier is very low.

3.1 Experimental Model

We generate time series data from polynomial models of the form

$$m(t) = p(t) + \epsilon \tag{3}$$

where ϵ is $N(0, \sigma)$ and σ is constant. We assume the polynomial is order 1, i.e.

$$p(t) = a + b \cdot t$$

The purpose of these experiments is to demonstrate the robustness in the presence of outliers of using a discretised time series rather than the the continuous data for clustering. Hence, we add a further term to Equation 3 to model the effect of outliers. A continuous time series is assumed to be generated by a sequence of observations from the model

$$m(t) = a + b \cdot t + \epsilon + r \tag{4}$$

where

$$r = s \cdot x \cdot y.$$

s is a constant, $x \in \{0, 1\}$ and $y \in \{-1, 1\}$ are observations of independent random variables, X and Y, where X has density

$$f(x) = p^x (1 - p)^{1-x}$$

and Y has density

$$f(y) = \frac{1}{2}.$$

r is a *random shock* effect that can occur with probability p, and if it occurs it has the effect of either adding or subtracting a constant s to the data (with equal probability).

A continuous time series is a sequence of observations from a model, now defined as

$$y(t) = p(t) + \epsilon + r \quad t = 1 \ldots n \tag{5}$$

A binary data series is generated by transforming a continuous series into series of above and below the median. If ϕ_y is the sample median of the data series $y(t), t = 1, \ldots, n$, then the associated discretised time series, z, is defined as

$$z(t) = \begin{cases} 1 \text{ if} & y(t) > \phi_y \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

A data set is parameterised as follows: there are k models of the form given in Equation 5, each of which generates l time series; each of the $l \cdot k$ time series is of length n and is sampled at the same points $t = 1, 2, \ldots, n$; σ defines the variability of static noise, s the level of random shocks and p the probability of shocks. From a data mining perspective, the clustering problem we are attempting to solve has the following properties:

- learning is unsupervised since cluster membership is not known *a priori*;
- cluster sizes are equal (l the same for all clusters k);
- there is no missing data (each series sampled at the same points);
- The number of clusters, k, is known *a priori*;
- the distribution of ϵ is constant for all observations and all series.

3.2 Experimental Procedure

We use the k-means algorithm with the correlation based distance metric given in Equation 2 for experimentation. We choose k-means as it is one of the most popular and simple clustering algorithms. Further experimentation will involve assessment of the clustering using alterative algorithms and distance metrics.

We initialise the centroids for k-means to a random data series. It is well known that k-means is sensitive to initial conditions [10], hence we repeat the classification algorithm with random initial conditions and then average over the runs. For any data set D of $l \cdot k$ time series derived from a particular set of k models, the clustering algorithm is run u times. For any particular parameter values, v different sets of k models are generated.

Clustering performance is measured by the classification accuracy, i.e. the ratio of the percentage of the data in the final clustering that is in the correct cluster. Note we are measuring accuracy on the training data rather than applying the data to a separate testing data set. We do this because wish to measure the effects of outliers in the training data rather than assess the algorithm's ability to solve the clustering problem. We use this measure rather than some of the alternatives (see [23]) since we know the correct clustering.

For a given clustering we measure the accuracy by forming a $k \times k$ contingency matrix. Since the clustering label may not coincide with the actual labelling (e.g. all those series in cluster 1 may be labelled cluster 2 by the clustering algorithm) we evaluate the accuracy (number correctly classified divided by the total number of series) for all possible $k!$ permutations of the columns of the contingency table. The achieved accuracy is the maximum accuracy over all permutations.

We average the accuracy over the u repetitions to find the average accuracy for a set of particular models, and average this data over the v different model sets to find the average performance for a particular set of parameter values. This average of averages we term *the average correct classification.*

All the parameters are given in Figure 1. Unless otherwise stated, th e parameter values used in all experimentation is given in brackets

Parameters	Meaning	Default value
experiment parameters		
k	Number of clusters	$k = 2$
n	Time series length	$n = 100$
l	Series per cluster	$l = 10$
u	Clusterings per model	$u = 20$
v	Number of models	$v = 20$
D	Data set consisting of $l \cdot k$ series	
m_i	A generating model	
model parameters		
$a_i, b_i, \ \ i = 1 \ldots k$	Linear parameters	
σ	Model noise	$\sigma = 10$
p	Outlier probability	
s	Random shock value	$s = 100$

Fig. 1. List of experimental parameters

3.3 Experimental Sequence

We demonstrate that the discretised data results in significantly better clusters when there are outliers in the data by conducting 4 experiments.

- Experiment 1 shows that treating a time series as a multivariate clustering problem can result in the failure to identify the correct underlying clustering (Section 3.4).
- Experiment 2 shows that discretising the data to above and below the median can mitigate against the effect of outliers for a particular model (Section 3.5).
- Experiment 3 shows that discretising the data does not significantly reduce the accuracy of a class of models when there are no outliers or outliers are very unlikely (Section 3.6).
- Experiment 4 shows that discretising the data does significantly increase the accuracy of a class of models when outliers are more likely (Section 3.7).

3.4 Experiment 1: Basic Linear Model

It has frequently been observed that clustering time series with vector based distance metrics (i.e. metrics that take no account of the ordering of the data) will result in degradation of performance. We demonstrate this by comparing performance of k-means using a Euclidean distance (Equation 1) with a correlation based metric (Equation 2) with data arising from the model given in Equation 3.1.

The parameters used were: $b_1 = 0.5$; $b_2 = -0.5$; a_1 and a_2 are uniformly sampled in the range $[100, 200]$ for each time series; and $p = 0$ (i.e. there are no random shocks).

Using Euclidean distance, the average correct classification was 78.45%, whereas the correlation based distance metric achieved 100% accuracy. This illustrates that the differing scales (different a_i values) can overwhelm the time based trends (fixed b_i values) and reinforces the point that vector based distance metrics can fail to detect regularities in time series data.

3.5 Experiment 2: Random Shocks Linear Model

This experiment shows that if there are outliers (random shocks) in the data then discretising can improve the accuracy of k-means clustering. Data was generated from the model described in Equation 5 then the discrete series were formed using Equation 6.

The parameters were as given in Figure 1 and in Section 3.4, except for the fact that p, the probability of a random perturbation of the series, may vary. Data was then generated with different values of p. Two example time series, one from each cluster, for $p = 0$ and $p = 0.5$ are shown in Figure 2. Figure 2 illustrates how the overlap between data increases with the increased chance of a random shock, and hence how the clustering task gets harder as p increases.

For each p value, $p = 0$ to 0.6 in increments of 0.01, 20 random models were generated (i.e. $v = 20$) and k-means was run for 20 times (i.e. $u = 20$) on data from each model. The average classification accuracy for each of the 61 different p values is shown in Figure 3 for both the continuous and discrete data with a random shock value of $s = 100$. Clearly the accuracy with the discretised data results in significantly higher accuracy even when the probability of a random shock is very low.

Figure 4 shows the even more dramatic effect on the accuracy of continuous data when the level of random shock is increased to 1000. As would be expected, increasing the level of random shock does not effect the accuracy of the clusters formed from the discretised data.

This experiment demonstrates the potential benefit of discretising the data for the model given. However, it would be of more interest to demonstrate the benefits over a class of models. The next two experiments address the issue of how discretising effects performance over a wider class of linear models than considered in Experiment 2.

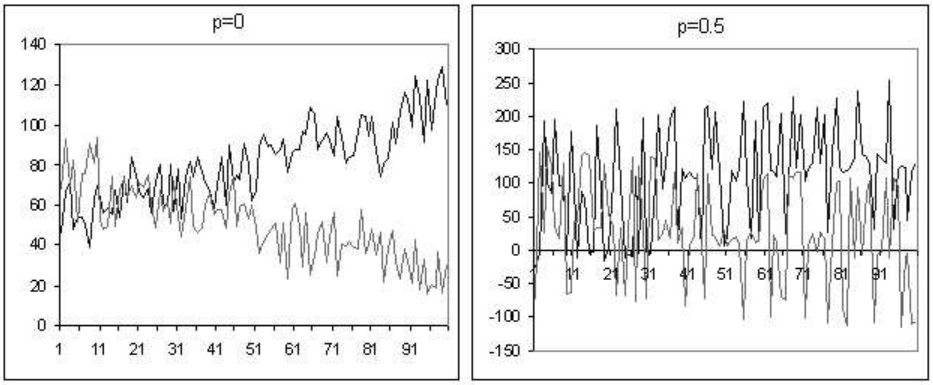

Fig. 2. Example data series on the left has no random shocks (p=0), the series on the right has an overwhelming number of random shocks (p=0.5)

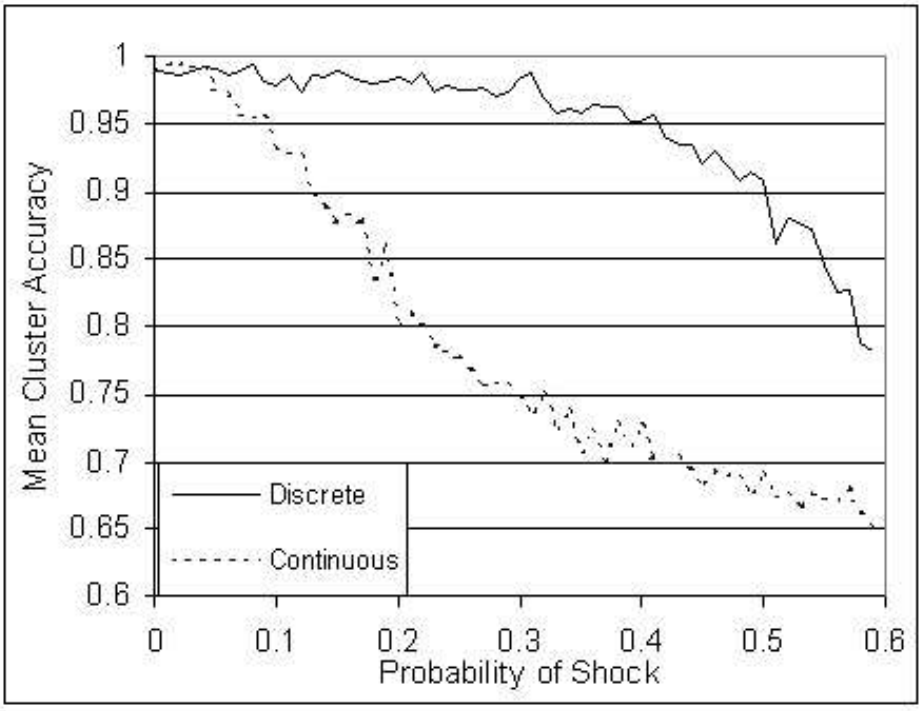

Fig. 3. The average classification accuracy of k-means when using continuous data and discrete data. p values in the range 0 to 0.6 and a random shock value s = 100

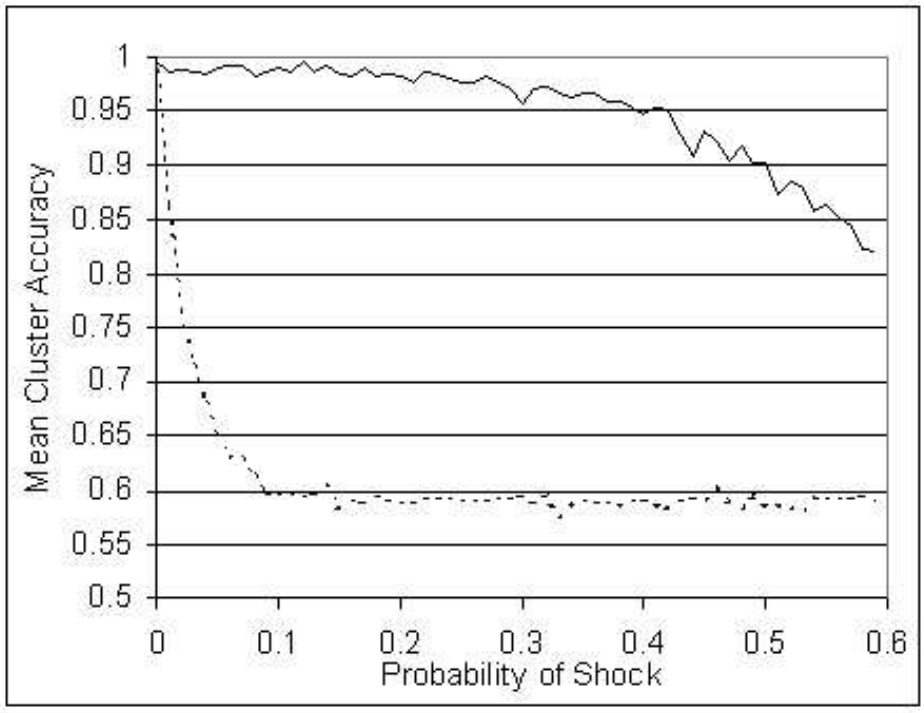

Fig. 4. The average classification accuracy of k-means when using continuous data (dotted line) and discrete data (solid line). p values in the range 0 to 0.6 and a random shock value value s = 1000

3.6 Experiment 3: Showing that using z does not significantly decrease accuracy

The objective of this experiment is to determine whether discretising the data significantly reduces the accuracy of the classification of the k-means algorithm using a correlation based distance metric. We perform this experiment with a sample from a wider class of models than used in Experiments 1 and 2. The format of the models is as given in Equations 5 and 6 and the default parameters are used (Figure 1).

Let M be the set of all models considered in the experiment, with an instance denoted m_i. Let C be the set $M \times M$ of generators of the two cluster model. ϕ_y is the population median of the average classification accuracy of the k-means algorithm (k known, random initial centroids) over the space of underlying models C and ϕ_z denotes the population median when using the discretised data. μ_y and μ_z are the associated population means. Given a random sample of model v, we wish to test $H_0 : \phi_z = \phi_y$ against the alternative $H_1 : \phi_z < \phi_y$ for a wider

class of models. For each model, b_i is now selected randomly on the interval $[-x, x]$, where x determines how likely given models are to be similar.

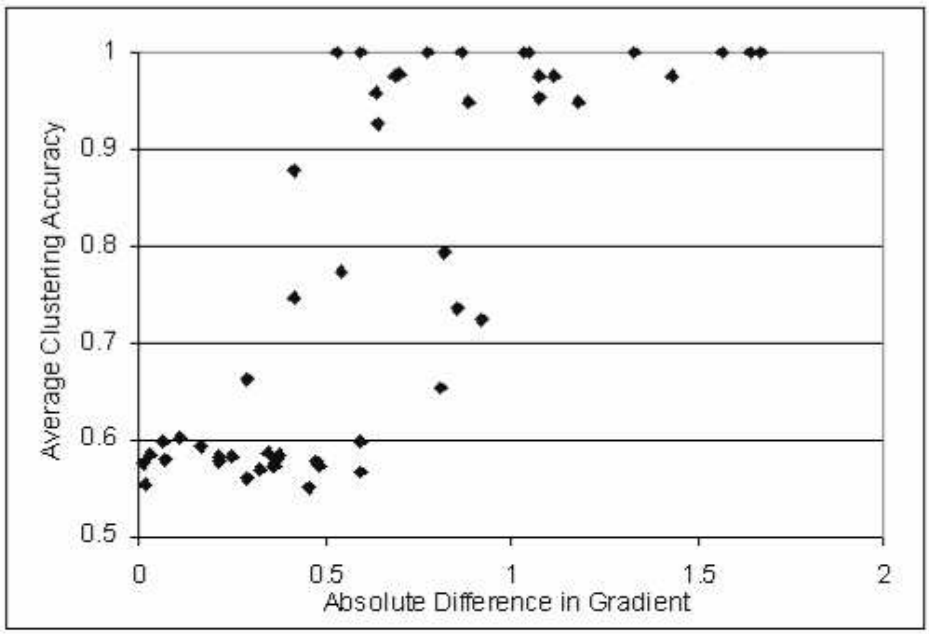

Fig. 5. Average clustering accuracy for varying difference of gradient between generating models using continuous data and no random shocks

Thus the gradients of the models may be close, making the clustering task very hard, or far enough apart to easily discriminate the series. Figure 5 shows how the accuracy of the k-means algorithm improves as the gap between the gradient of the generating models increases when there are no random shocks. The accuracy data appears to be grouped into three clusters. When the difference is below approximately 0.4 the algorithm is unable to properly distinguish the clusters. When the gradient is above 1 the algorithm is almost always completely accurate. In the region between k-means can have a wide range of accuracy. The accuracy when using the discrete data shows a similar pattern. In order to compare performance, we concentrate on the types of model where differences in accuracy are most likely to be caused by the transformation of the data rather than on poor initial conditions for the clustering algorithm, hence we restrict the gradient range to $[-0.5, 0.5]$ (i.e. $x = 0.5$).

To reverify that this restriction is not masking a difference in performance on discrete and continuous data, we examined the difference in accuracy using paired samples (i.e. we evaluate k-means accuracy with the continuous and the

discrete data generated from the same underlying models) and a sample size of $v = 50$ for a range of fixed gradients. The results, shown in Figure 6, suggest there is little change in distribution, indicating that the effect of the discretisation is independent of the difficulty of the classification problem for gradients less than 1.

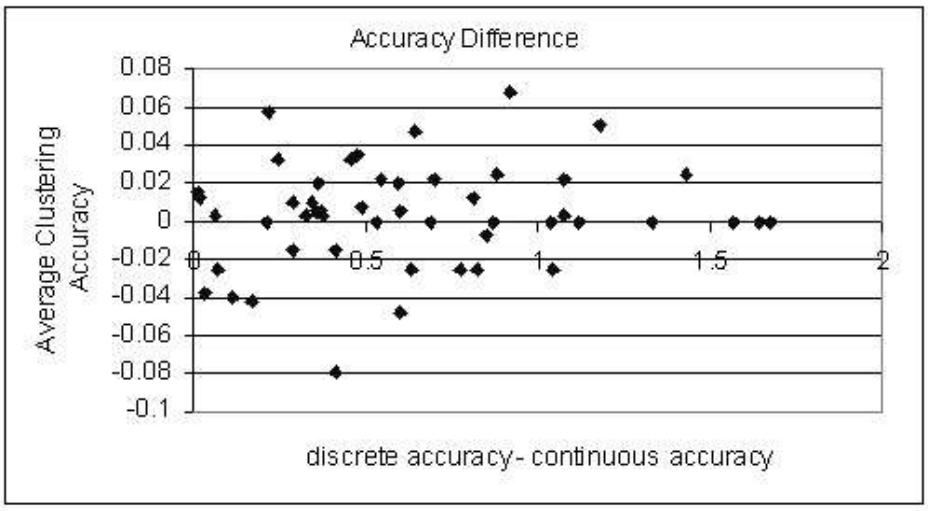

Fig. 6. Difference in average clustering accuracy against difference in gradient of the underlying models. A positive data indicates the clustering accuracy on discrete data was higher than that on continuous data

Table 1 summarises the results when the experiment was repeated with $x = 0.5$. Of the 50 trials, there were 24 trials with a positive difference (i.e. continuous

Table 1. Clustering accuracy summary for paired samples. The difference series is the continuous data minus the discrete data

	Mean	Median	Min	Max	StDev
Continuous	80.13%	91.15%	56.60%	100%	17.80
Discrete	79.84%	86.30%	57.80%	100%	17.64
Difference	2.67%	0%	-5%	8.7%	2.67

data resulted in a higher accuracy than the discrete data), 4 had no difference and 21 had negative difference. There is a positive mean difference but the median difference is zero and we cannot reject the null hypothesis that $H_0 : \phi_z = \phi_y$ for the alternative $H_1 : \phi_z < \phi_y$ using the Wilcoxon's test for matched pairs.

It is worth noting that we cannot rejected the hyptothesis $H_0 : \mu_z = \mu_y$ in favour of the alternative $H_1 : \mu_z < \mu_y$ using a t-test. Despite this result, we use non-parametric tests due to the decidely non normal nature of the data.

To verify there is in fact no significant difference in the median clustering accuracy for the models considered, we re-ran the experiment with unmatched pairs and 100 models in each sample ($v = 100$).

Table 2. Accuracy summary for unmatched models

	Mean	Median	Min	Max	StDev
Continuous	76.801%	73.00%	56.20%	100%	16.96
Discrete	76.798%	72.85%	55.50%	100%	15.83

Table 2 summarises the results. The difference in the mean is neglible, and using the Mann-Whittley test we cannot reject the null hypothesis $H_0 : \phi_z = \phi_y$ in favour of the alternative $H_1 : \phi_z \neq \phi_y$.

These results clearly demonstrate that discretising the data does not decrease the accuracy of the k-means clustering algorithm used to cluster data derived from two models of the form given in Section 3.1 when there are no outliers in the data. The next experiment shows that the accuracy of the clustering significantly improves even when the probability of an outlier is very small.

3.7 Experiment 4: Showing Using a discretised series increases accuracy

To demonstrate the desirability of discretising, we repeat experiment 3 for various values of p with both paired and unpaired samples. All other parameters are identical to those used in results presented in Section 3.6 ($v = 50, x = 0.5$). Figure 3.7 shows how the accuracy difference changes as the probability of an outlier increases. Each data represents the median of 50 evaluations, where each evaluation consists of 20 runs of the k-means algorithm. There is an initial dramatic decrease in accuracy of clustering using the continuous data. As the probability of an outlier increases the accuracy difference between using discretised and continuous data decreases. This is because the noise eventually overwhelms the algorithms ability to cluster correctly.

To illustrate the effect of outliers more clearly, Figure 8 shows the results for the same experiment using a smaller range of p. Clearly the clustering algorithm is performing much better with the discretised data even when the probability of outlier is very low.

Figure 9 shows a repeat of the experiment described by Figure 8 with unpaired samples. For contrast with Figure 8, the mean values rather than the medians are shown, but the pattern in both averages is the same. A very small probability of outliers results in a much improved performance when the discretised data is used.

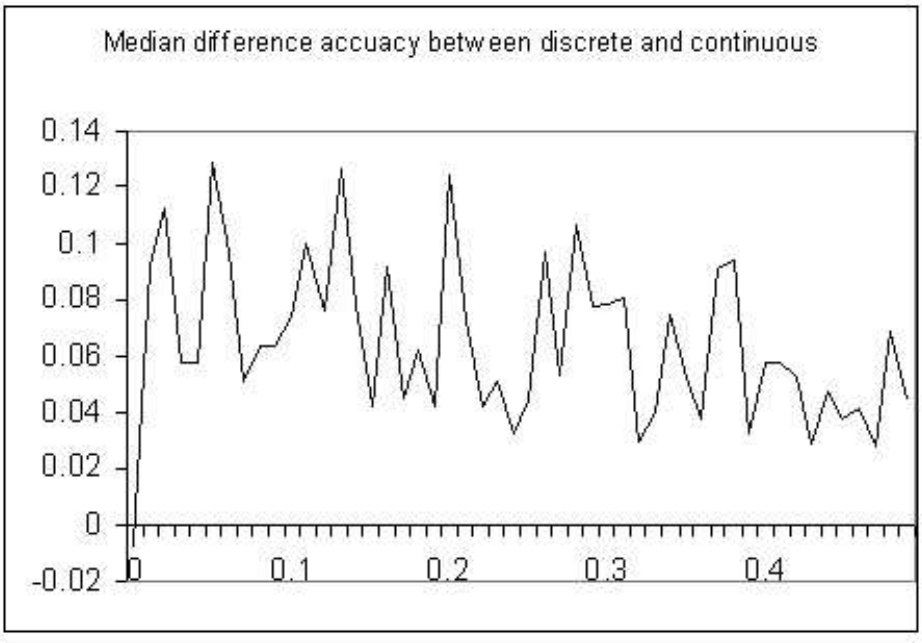

Fig. 7. Difference in clustering accuracy for probability of an outlier between 0 and 0.5 with paired samples

Finally, to emphasise the point further, we fixed the number of outliers so that each series of 100 data had exactly one outlier and repeated the paired experiment. Of the 50 trials, 9 resulted in the continuous data having higher accuracy, in 1 trial the accuracy was the same and in the remaining 40 the accuracy was greater when the discrete data was used.

Using Wilcoxon's signed-rank test for matched pairs, the null hypothesis $H_0 : \phi_d = 0$ can be rejected in favour of the alternative $H_1 : \phi_d < 0$ at the 1% level

4 Conclusions and Future Direction

The clustering of time series is a field that has attracted the interest of researchers from a wide range of disciplines. This report has provided a brief review of the techniques used, including a description of the types of models assumed, the distance metrics employed and the clustering techniques used. Many real world time series have the unfortunate property that they contain outliers, and the aim of this research is to demonstrate that if discretised series are used instead of the continuous data then the effect of outliers can be lessened significantly.

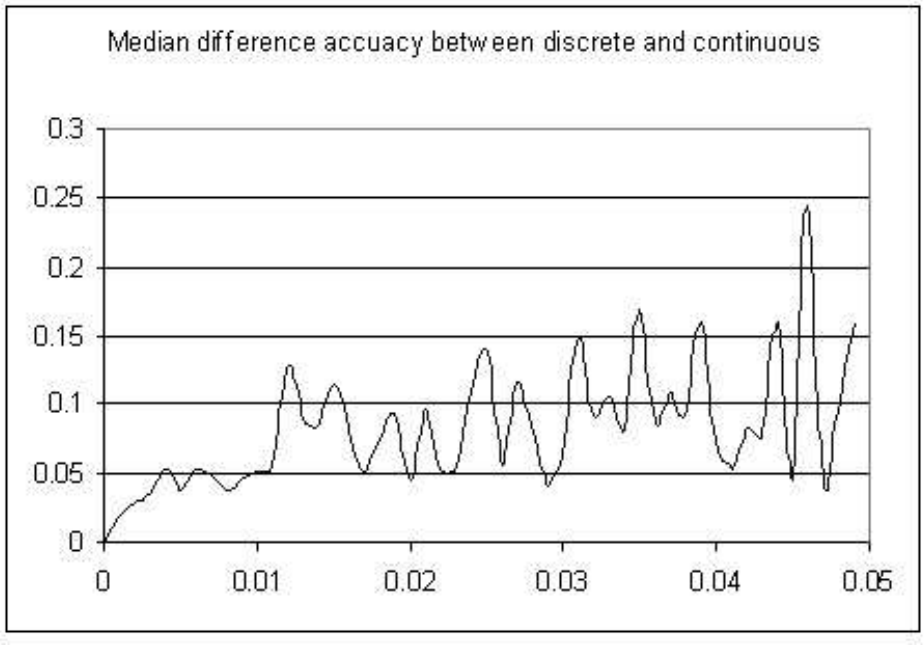

Fig. 8. Difference in clustering accuracy for probability of an outlier between 0 and 0.05 with paired data

We have demonstrated how, for a certain class of model, distance metric and clustering algorithm, discretising the series into binary series of above and below the median can improve the clustering accuracy when there are outliers in the data, even when the probability of an outlier is very small.

Although there are benefits from using the binary series of above and below the median when there are outliers, it obviously means some of the information in the original data is discarded. It is worthwhile discovering how much this effects the quality of clusters formed.

The obvious way of extending this work would be to assess the effect of discretisation when the data arises from other models and when alternative distance metrics and/or clustering algorithms are employed. It is also a logical extension to apply the technique to real world data.

Working with binary series can often allow for significant speed improvements with model fitting and clustering techniques, and this could be another benefit of discretisation.

References

1. Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient Similarity Search In Sequence Databases. In D. Lomet, editor, *Proceedings of the 4th Inter-*

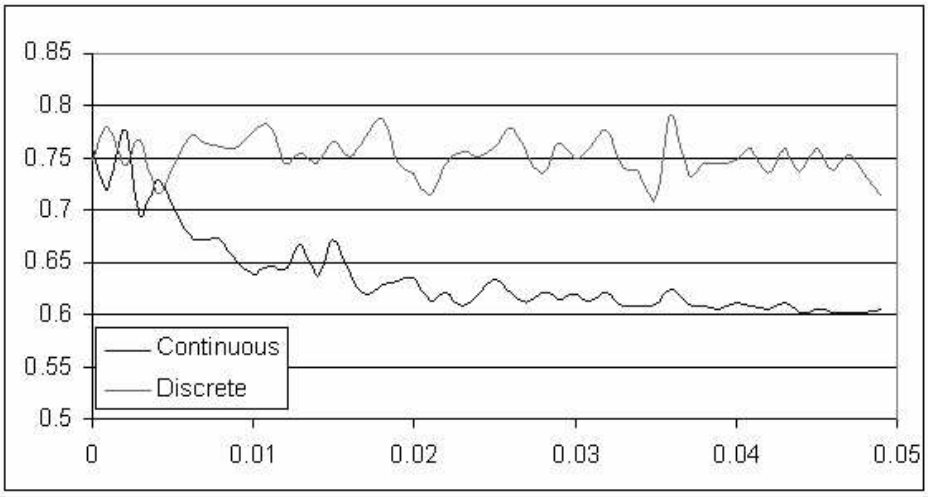

Fig. 9. Difference in clustering accuracy for probability of an outlier between 0 and 0.05 with unpaired data

national Conference of Foundations of Data Organization and Algorithms (FODO), pages 69–84, Chicago, Illinois, 1993. Springer Verlag.

2. Jonathan Alon, Stan Sclaroff, George Kollios, and Vladimir Pavlovic. Discovering clusters in motion time-series data. In *IEEE Computer Vision and Pattern Recognition Conference (CVPR)*, 2003.
3. A. J. Bagnall and I. Toft. An agent model for first price and second price private value auctions. In *to appear in Proceedings of the 6th International Conference on Artificial Evolution*, 2003.
4. A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago, April 2001.*, 2001.
5. Z. Bar-Joseph, G. Gerber, D. Gifford, T. Jaakkola, and I. Simon. A new approach to analyzing gene expression time series data. In *Proceedings of The Sixth Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 39–48, 2002.
6. Roberto Baragona. A simulation study on clustering time series with metaheuristic methods. *Quaderni di Statistica*, 3, 2001.
7. R. Blender, K. Fraedrich, and F. Lunkeit. Identification of cyclone-track regimes in the north atlantic. *Quart J. Royal Meteor. Soc.*, (123):727–741, 1997.
8. Z. Bohte, D. Cepar, and K. Kosmelj. Clustering of time series. In *Proceedings in Computational Statistics*. Physica-Verlag, 1980.
9. G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control, 3rd Edition*. Prentice Hall, 1994.
10. Paul S. Bradley and Usama M. Fayyad. Refining initial points for K-Means clustering. In *Proc. 15th International Conf. on Machine Learning*, pages 91–99. Morgan Kaufmann, San Francisco, CA, 1998.

11. Igor V. Cadez, Scott Gaffney, and Padhraic Smyth. A general probabilistic framework for clustering individuals and objects. In *Knowledge Discovery and Data Mining*, pages 140–149, 2000.
12. Igor V. Cadez, David Heckerman, Christopher Meek, Padhraic Smyth, and Steven White. Visualization of navigation patterns on a web site using model-based clustering. In *Knowledge Discovery and Data Mining*, pages 280–284, 2000.
13. Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Finding similar time series. In *Principles of Data Mining and Knowledge Discovery*, pages 88–100, 1997.
14. Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *Knowledge Discovery and Data Mining*, pages 16–22, 1998.
15. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data. *J. R. Stat. Soc. B*, 39:1–38, 1972.
16. Evangelos Dermatas and George Kokkinakis. Algorithm for clustering continuous density HMM by recognition error. *IEEE Tr. On Speech and Audio Processing*, 4(3):231–234, 1996.
17. Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.
18. Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings 1994 ACM SIGMOD Conference, Mineapolis, MN*, pages 419–429, 1994.
19. Sergio M. Focardi. Clustering economic and financial time series: exploring the existence of stable correlation conditions. Technical Report 2001-04, The Intertek Group, 2001.
20. Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. Technical Report 99-15, Department of Information and Computer Science, University of California, 1999.
21. Scott Gaffney and Padhraic Smyth. Curve clustering with random effects regression mixtures. In C. M. Bishop and B. J. Frey, editors, *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.
22. Amir B. Geva and Dan H. Kerem. *Fuzzy and Neuro-Fuzzy Systems in Medicine*, chapter 3. Brain state identification and forecasting of acute pathology using unsupervised fuzzy clustering of EEG temporal patterns. CRC Press, 1998.
23. Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.
24. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, New Jersey, 1988.
25. E. J. Keogh and M. J. Pazzani. A simple dimensionality reduction technique for fast similarity search in large time series databases. In T. Terano, H. Liu, and A. Chen, editors, *Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference, PAKDD 2000*, volume 1805, pages 122–133, Kyoto, Japan, 2000. Springer.
26. Eamonn Keogh, Jessica Lin, and Wagner Truppel. Clustering of streaming time series is meaningless. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003.
27. Filipp Korkmazskiy, Biing-Hwang Juang, and Frank Soong. Generalized mixture of HMMs for continuous speech recognition. In *Proceedings IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 2, pages 1443–1446, 1997.

28. K. Kosmelj and V. Batagelj. Cross-sectional approach for clustering time varying data. *Journal of Classification*, 7:99–109, 1990.
29. Cen Li and Gautam Biswas. Clustering sequence data using hidden markov model representation. In *SPIE'99 Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology*, pages 14–21, 1999.
30. Cin Li. *A Bayesian approach to temporal data clustering using the hidden Markov model methodology*. PhD thesis, Vanderbilt University, Nashville, 2000.
31. Cin Li and Gautam Biswas. Profiling of dynamic system behaviors using hidden markov model representation. In *Proceedings of the ICSC'99 Advances in Intelligent Data Analysis(AIDA'99)*, 1999.
32. Cin Li and Gautam Biswas. Temporal pattern generation using hidden markov model based unsupervised classification. In D. Hand, K. Kok, , and M. Berthold, editors, *Advances in Intelligent Data Analysis, Lecture Notes in Computer Science vol. 1642*. Springer, 1999.
33. Cin Li and Gautam Biswas. Bayesian clustering for temporal data using hidden markov model representation. In *proceedings of the Seventeenth International Conference on Machine Learning*, pages 543–550, 2000.
34. J. MacQueen. Some methods for classification and analysis of multivariate observations. In Lucien M. Le Cam and Jerzy Neyman, editors, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I,Statistics*. University of California Press, 1967.
35. Elizabeth Ann Maharaj. A significance test for classifying arma models. *Journal of Statistical Computation and Simulation*, 54:305–331, 1996.
36. Elizabeth Ann Maharaj. Clusters of time series. *Journal of Classification*, 17:297–314, 2000.
37. Eamonn Keogh Michail Vlachos, Jessica Lin and Dimitrios Gunopulos. A wavelet-based anytime algorithm for k-means clustering of time series, 2003.
38. Tim Oates. Identifying distinctive subsequences in multivariate time series by clustering. In S. Chaudhuri and D. Madigan, editors, *Fifth International Conference on Knowledge Discovery and Data Mining*, pages 322–326, San Diego, CA, USA, 1999. ACM Press.
39. Tim Oates, Laura Firoiu, and Paul Cohen. Clustering time series with hidden markov models and dynamic time warping. In *Proceedings of the IJCAI-99 Workshop on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21, 1999.
40. Tim Oates, Laura Firoiu, and Paul R. Cohen. Using dynamic time warping to bootstrap HMM-based clustering of time series. *Lecture Notes in Computer Science*, 1828:35–52, 2001.
41. Tim Oates, Matthew D. Schmill, and Paul R. Cohen. Identifying qualitatively different outcomes of actions: Gaining autonomy through learning. In Carles Sierra, Maria Gini, and Jeffrey S. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents*, pages 110–111, Barcelona, Catalonia, Spain, 2000. ACM Press.
42. Paul Ormerod and Craig Mounfield. Localised structures in the temporal evolution of asset prices. In *New Approaches to Financial Economics*. Santa Fe Conference, 2000.
43. Domenico Piccolo. A distance measure for classifying ARIMA models. *Journal of Time Series Analysis*, 11(2):153–164, 1990.
44. Marco Ramoni, Paola Sebastiani, and Paul Cohen. Bayesian clustering by dynamics. *Machine Learning*, 47(1):91–121, 2002.

45. Greg Ridgeway. Finite discrete Markov processes. Technical Report MSR-TR-97-24, Microsoft Research, 1997.
46. Padhraic Smyth. Clustering sequences with hidden markov models. In Michael C. Mozer, Michael I. Jordan, and Thomas Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 648. The MIT Press, 1997.
47. P. Tong and H. Dabas. Cluster of time series models: An example. *Journal of Applied Statistics*, 17:187–198, 1990.
48. Shi Zhong. *Probabilistic model-based clustering of complex data*. PhD thesis, University of Texas at Austin, 2002.
49. Shi Zhong and Joydeep Ghosh. HMMs and coupled HMMs for multi-channel EEG classification. In *Proc. IEEE Int. Joint Conf. on Neural Networks*, 2002.
50. Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering. In *Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE)*, 2002.
51. Shi Zhong and Joydeep Ghosh. A unified framework for model-based clustering and its application to clustering time sequences. Technical report, Department of Electrical and Computer Engineering, University of Texas, 2002.
52. Shi Zhong and Joydeep Ghosh. Scalable, balanced model-based clustering. In *Proceedings of SIAM Int. Conf. on Data Mining*, 2003.